Spring 5-1-2002

A Fantasy Football Web Application with Active Server Pages and Access Database

Jeremy Pauli
Dakota State University

A Fantasy Football Web Application with Active Server Pages and Access Database

By: Jeremy Pauli

A project submitted in partial fulfillment of the requirements for the
Master of Science in Information Systems

Dakota State University
2002

Abstract

This MSIS project will consist of building a full-service fantasy football website, utilizing dynamically created webpages that will allow participants to take part in a non-traditional fantasy football league. Fantasy Football is a game where participants select or draft a predetermined number of players from NFL rosters to compete against other participants. Scoring is determined by the game performance of players where touchdowns, receiving yards, and other statistics are turned into points. The participant with the most points wins. The project will be implemented on a 1.8GHz Dell with 256 RAM running Internet Information Services 5.1 (IIS) on Windows XP Professional Edition with Office XP. The objectives of the project are to create an online fantasy football website, learn more about ASP, CSS and Access, and to learn more about the planning necessary to successfully manage large projects.

Table of Contents

Project Approval Form..... ii

Abstract..... i

Introduction..... 1

Statement of problem..... 9

Objectives..... 10

Scope of the Project..... 13

Results and Conclusions..... 15

References..... 20

Appendices

Appendix A: Application Screen Shots
Appendix B: Database Design, Schema & Administration
Appendix C: Work Breakdown structure
Appendix D: Gantt Chart
Appendix E: Project Source Code (.html, .asp & .css) & Access Files(.mdb) CD

Figures

Figure 1..... 3
Figure 2..... 4
Figure 3..... 4
Figure 4..... 5
Figure 5..... 5
Figure 6..... 8

Tables

Table 1..... 6
Table 2..... 6
Table 3..... 6

Introduction of the project

This MSIS project will consist of building a full-service fantasy football website, utilizing dynamically created webpages that will allow participants to take part in a non-traditional fantasy football league. (NOTE: the term 'participants' will be used to describe the users of the league and the term 'players' will be used to describe actual NFL players)

Fantasy Football is a game where participants select or draft a predetermined number of players from NFL rosters to compete against other participants. Scoring is determined by the game performance of players where touchdowns, receiving yards, and other statistics are turned into points. The participant with the most points wins.

The league will allow each participant to create a team, change rosters on a weekly basis and monitor their results against other participants. Participants will also be allowed to form 'divisions' of teams to compete against a smaller group of participants rather than competing against all participants in the league. Participant standings can be figured based on the entire league and by each individual division.

There are currently many fantasy football games available on the web but they are all in, what I call, the 'traditional' style. That is, they require participants to form their own leagues or divisions of friends, and then all at once they must draft players to fill their rosters. Usually, this draft consists of drafting 20-35 players per person and can be very time consuming and confusing to non-hardcore football fans. Each week in the traditional style of fantasy football, participants pair off against one-another in a head-to-head match up, and whichever participant scores more points from their combined roster will win the match up and get credit for a win. Therefore, wins and losses determine league standings, not player yards, touchdowns, field goals or total points.

The website developed in this project will allow participants to sign up individually, create a team each week from the entire roster of all NFL teams or to keep basically the same team each week only removing players if they are hurt or have a bye week. The design of this type of league is much simpler to understand from the participant's perspective but is not without the high level of strategy that is found in the traditional leagues. The diagrams following contrast the traditional method versus my method:

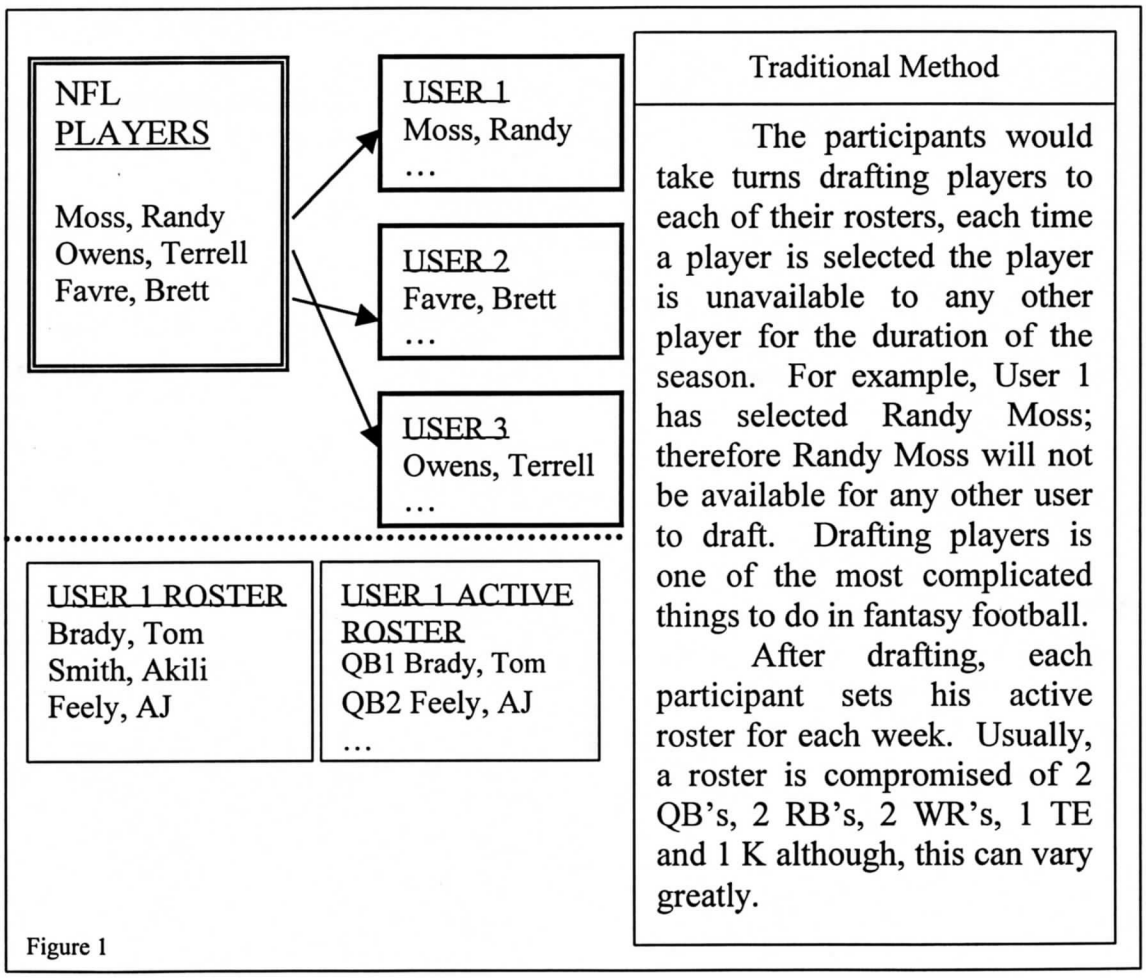


Figure 1

MY DYNAMIC ROSTER METHOD:

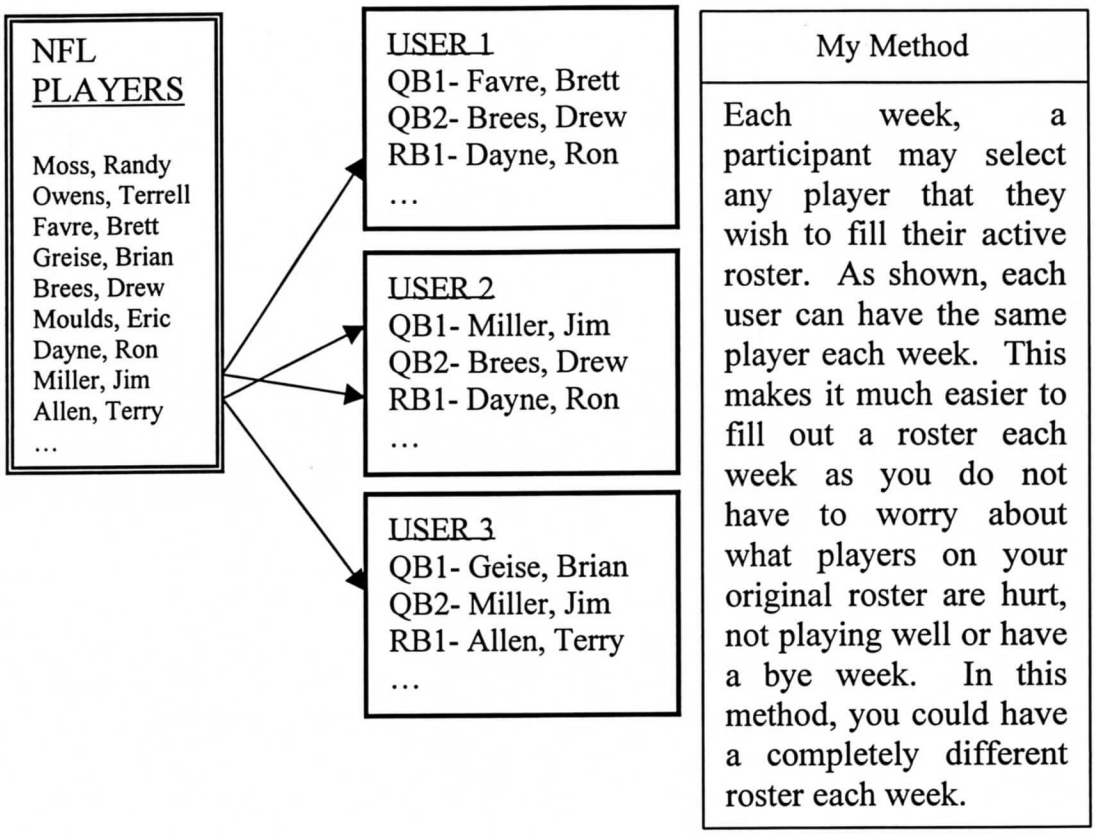


Figure 2:

TRADITIONAL WEEKLY SCORING METHOD:

Week 1 Schedule
User 1 vs. User 2
User 3 vs. User 4

User 1	**User 2**	**User 3**	**User 4**
58 points	54 points	42 points	46 points

RESULT
User 1: 1 Win – 0 Loss
User 2: 0 Win – 1 Loss

RESULT
User 3: 0 Win – 1 Loss
User 4: 1 Win – 0 Loss

WEEK 1 STANDINGS
1. User 1: 1 Win – 0 Loss
1. User 4: 1 Win – 0 Loss
2. User 3: 0 Win – 1 Loss
2. User 2: 0 Win – 1 Loss

The traditional method pits participants against each other each week. In this example User 1 competes against User 2 and User 3 against User 4. As you can see User 1 and User 4 are victorious and the weekly standings at the bottom reflect the week's competition.

You may also notice that even though User 2 scored more points than User 4, User 2 is still behind User 4 in the standings.

Figure 3:

MY WEEKLY SCORING METHOD:

WEEK 1

User 1	**User 2**	**User 3**	**User 4**
58 points	54 points	42 points	46 points

WEEK 1 STANDINGS

1. User 1 – 58 Points
2. User 2 – 54 Points
3. User 4 – 46 Points
4. User 3 – 42 Points

My method ranks participants based on the number of points earned each week, not on the outcome of a head-to-head match up. This method offers several benefits:

First, participants will not get discouraged if they lose their first few head-to-head match-ups because they are always able to have one big week and get right back into it.

Second, there is less strategy involved; you don't have to worry as much about who your opponent has on their active roster. Is less strategy good?

Figure 4:

OVERALL APPLICATION FLOW:

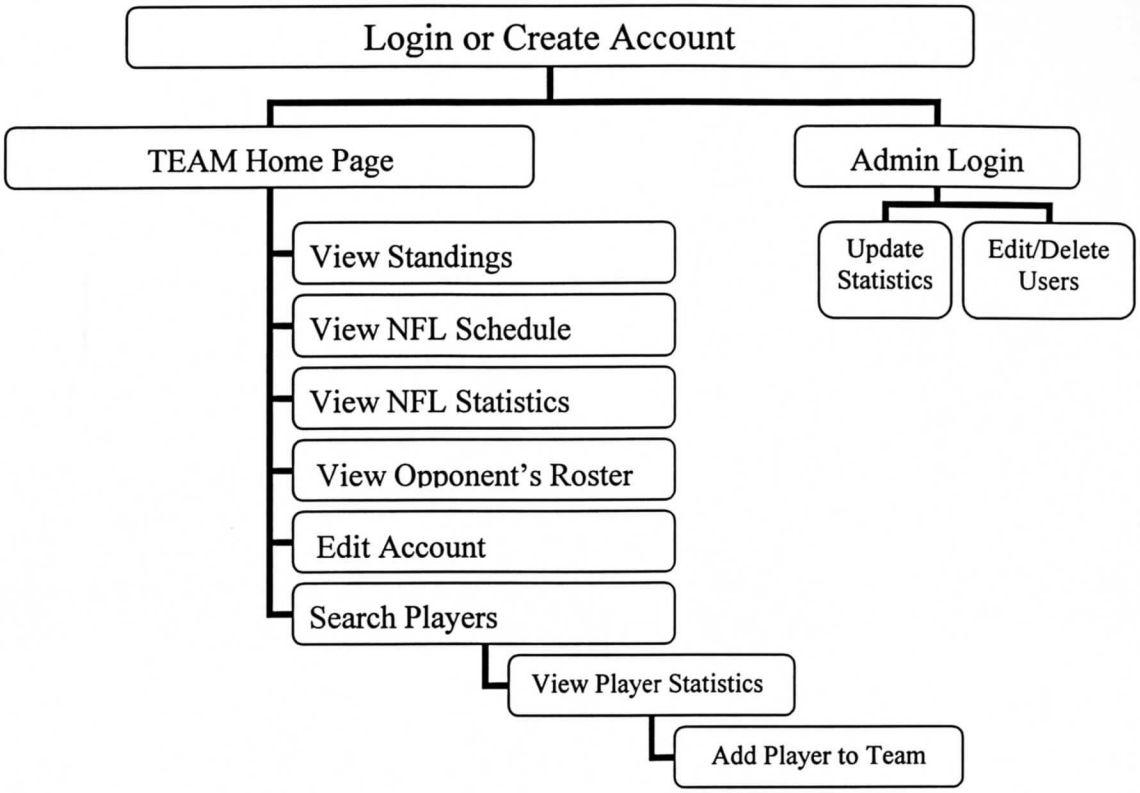


Figure 5:

Scoring Methods:

Generally, in fantasy football, points are awarded for performance in many categories ranging from touchdowns to tackles. In this project, points will be awarded on the following basis:

Quarterback Scoring	
Category	**Points**
1 Pass yard	1
1 Pass TD	50
1 Rush Yard	1.5
1 Rush TD	75
1 Interception	-50
1 Fumble Lost	-50

Table 1

Running Back & Wide Receiver Scoring	
Category	**Points**
1 Pass yard	1
1 Pass TD	50
1 Rush Yard	1.5
1 Rush TD	75
1 Rec. Yards	2
1 Rec. TD	75
1 Fumble Lost	-50

Table 2

Kicker Scoring	
Category	**Points**
1 Extra Point Made	10
1 Field Goal Made	25
1 Extra Missed	-10
1 Field Goal Missed	-25

Table 3

Technical Specifications:

The project will be implemented on a 1.8GHz Dell with 256 RAM running Internet Information Services 5.1 (IIS) on Windows XP Professional Edition with Office XP. IIS is very scalable server architecture capable of handling million of hits a day but still easy to use due to its full integration into Windows 2000 based operating systems. What does this 'integration' mean? IIS allows the use of Active Server Pages for the construction of web applications. Web applications are protected with IIS's ability to run applications separate from core IIS processes which increases stability as well as having the ability to monitor and tune the amount of CPU cycles a specific application process is assigned (Internet Information Services Features). This protection allows IIS to be stable because the separation of application processes and core processes allows the core process to use the required system resources to continue running.

IIS is easily installed by simply choosing 'Add Programs' from the 'Control Panel' menu and selecting the 'IIS' Windows component. Administration is also easy

with the use of the Microsoft Management Control or MMC Snapin. The only change necessary for the project was the installation of the FrontPage Server Extensions Snapin. The installation process of the FrontPage Server Extensions is also very easily done inside the MMC.

The database connection will be an OLE DB connection. OLE DB connections are "a specification for a set of data access interfaces designed to enable a multitude of data stores, of all types and sizes, to work seamlessly together" (Microsoft OLE DB White Paper).

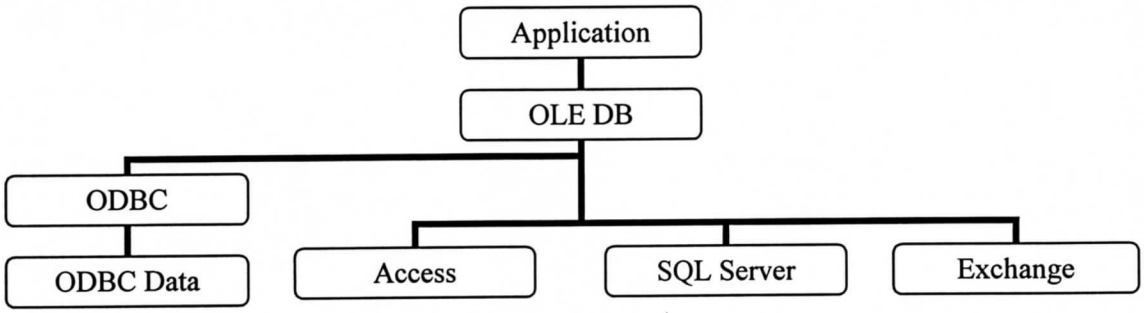


Figure 6

OLE DB has many advantages: First, OLE DB can access all types of data stores for great flexibility across different platforms. Second, OLE DB is able to provide an object oriented approach to accessing all OLE data objects with ADO. Finally, OLE DB is generally thought to be faster and easier to use than ODBC (Buser, et al.). One final interesting note is that OLE DB is the only method that currently allows you to query an Exchange email store (Microsoft OLE DB White Paper).

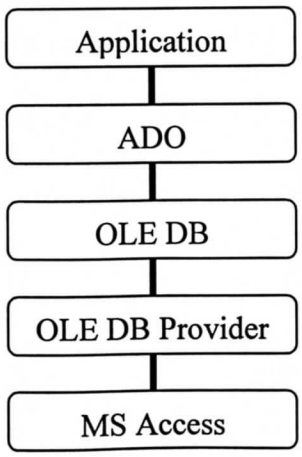


Figure 7

The final piece of the DB connection is the ADO level or ActiveX Data Object. The ADO acts as an application programming interface between the ASP code application and the OLE DB. (See Figure 7) ADO is a set of objects contained in the Microsoft Data Access Components (MDAC) that allow us to program data access logic from ASP rather than directly interfacing with the OLE DB which is much more complex than ADO (Buser, et al.).

Database Optimization:

The Access database used for this project was optimized for performance by using the 'Performance Analyzer' inside of Access (See AppendixB-13 for a detailed description of this process) to help decrease the time for page loads. The method that Access uses for optimizing a database is to analyze the database structure and determine which fields should be indexed. This indexing method is used because Access cannot change the physical storage method of the database. Indexes are access structures that speed up the retrieval of records in response to a query. Indexes can be built on individual or a group of fields in a table and there are several different types of indexes, such as, primary, clustering and secondary (Elmasri & Navathe).

Indexes speed up a search because they allow the CPU to read in more values for search criteria than reading from the actual data table. For example, if a CPU can read in one block of data at a time and one block of data contains 5 records, the CPU can only look at those 5 records to determine if they meet the criteria. On the other hand, the index is a much smaller table containing only the index field value and a record pointer

and so one block of an index table can contain 25 field values. Using this index, the CPU can determine which records meet the selection criteria much faster due to the decreased number of memory accesses required.

Microsoft Access is the database platform chosen for this project because of its ease of use for the league administrator or commissioner. Access is a very easy platform to use because all of interfaces used are visual and Access makes the data import from Yahoo.com very easy. Also, with the small number of records that are necessary for the fantasy football league a bigger database package would be unnecessary and perhaps an overkill.

Statement of the problem

How is the problem being addressed now?

Fantasy football league software is widely available in many different forms. A Google.com search (January, 2002) for 'fantasy football software' returned about 92,600 hits. A large number of the hits were for software packages that can manage a league, meaning that the software will simply automate the normal task of figuring points and standings for each user each week. Very few of the packages are an all-in-one solution with management functionality, along with the dynamic creation of web pages, and a completely web based user interface.

The problem is the same as described in the project proposal. No changes where made to the project plan with regards to the type of problem being addressed.

One of the most challenging parts of the project was actually the search for the most current and effective technology that is being used for fantasy football software. I

emailed several of the main competitors in the online fantasy football market (ESPN.com, WallStretSports.com, Smallworld.com, etc.) and none of the companies would tell me what types of technologies they use. This has made finding detailed information about the most current technology very hard. To make up for this lack of knowledge, I will assume that my solution of using Active Server pages and Access Database is current and robust enough to handle a fantasy football website.

Objectives

1. Create an online fantasy football website
2. Learn more about ASP, CSS and Access
3. Learn more about the planning necessary to undertake large projects

Smallworld.com (now part of The Sporting News) is the best on-line fantasy football game that I have found. My application and the Smallworld application are similar in a few ways, both applications are entirely web based and both share a similar scoring method. But there are several differences (or improvements, as I like to call them):

1. My application is much simpler to use and understand. The Smallworld method uses a larger roster consisting of 10-players, a salary cap method to limit the 'quality' of the players and a limited number of roster moves each week. My application utilizes a 6-player roster with no salary cap. The salary cap for the Smallworld method assigns a dollar value to each player in order

to limit the number of top-shelf players that can be on a participant's roster. I see several problems with this:

- One, managing the salary cap and roster moves can be tricky for the participants. For example, you can spend all of your 'money' and still have one player to purchase. This results in an incomplete roster and a score of zero for the week.
- Second, the salary cap method and large roster size forces participants to search for 'bargain' players that can be confusing or even tedious for less intense participants.
- And finally, having only a limited number of roster moves can severely limit the participants' ability to have a full roster with bye weeks and injuries to players. This again, can cause a user to have a roster position that does not score any points. Overall, the proposed method will be much simpler but will still involve a large amount of strategy with an easy-to-use interface.

2. Managing NFL Statistics. One of the most daunting aspects of creating a fantasy football application is the management of the almost 800 players in the NFL that could be selected to a user's roster. To solve this dilemma, I have chosen to download free stats from Yahoo.com. Yahoo allows private parties to download the current stats as well as the previous week's stats. The stats are downloaded as text files and then imported into Access with almost no alterations. This eliminates the many hours of data entry that would have been required if the downloadable stats were not present. This is a huge advantage over the

Smallworld method because generally in Smallworld, players are only available after they have scored for a week or two. Utilizing the statistics download makes the entire roster of NFL players available to each user. In fact, the Yahoo statistics download even contains individual statistics for defensive players and punters.

Deliverables

The main deliverable is a web application that allows participants to compete against one another in a fantasy football league with many divisions. The web application also features the complete and up-to-date list of players from all teams. Of course, the project also includes the required documentation for installation and operation of the application.

Scope of the Project:

The project was completed by following the activities in the Gantt Chart (Appendix E) and the Work Breakdown Structure (Appendix D). The project followed the general phases of Planning, Analysis, Design and Implementation.

In the planning phase I researched the current fantasy football websites and software to find out what types of league software is currently available. This search indicates that many different types of leagues are being commercially offered, as well as some league being offered free.

From this research, I began to form a list of basic requirements that would make a good league. I also identified several features that I thought would give my league an advantage over other leagues that were available. The factors were:

- Ease of use: Many of the leagues that I found were not easy to use or assumed a lot prior knowledge about football and other fantasy football leagues. I wanted my league to be easy to use for people new to fantasy football and new to web interfaces as well.
- Ease of Administration: Most leagues require a great deal of time and effort from the administrator or commissioner as they are usually known as. I tried to design the league to make it easy for a non-technical person to take control of the administration duties. All of the administration duties can be done through a web interface except for the statistics import and conversion that is done in Access and even this process is heavily automated with the help of a Microsoft Wizard and the Import Guide (Appendix C).
- Up to date NFL roster and statistics information: The use of a statistics download not only greatly reduces the amount of administration time needed by eliminating data entry but also ensures that the latest roster information is available to allow the greatest flexibility for user when they select their individual rosters.
- Flexible Scoring system: The league administrator can at any time change the scoring variables to suit the desires of the rest of the league members.

This is a major improvement over the majority of leagues that are offered. Most have very little flexibility in the scoring systems.

- Ease of including new functionality: The league is designed so that additional functionality is easy to build in. Whether it is including defensive players in the roster or creating multiple layers of divisions or password protecting the divisions, it is easy to add on.

After the planning was complete, I moved on the analysis phase. First, I planned the overall application flow (See Figure 5). Next, I looked at the basic structure of the database. This involved determining what fields would be needed and what data types would be appropriate. More information about the database can be seen in Appendix B.

After the analysis phase was complete I moved on the design phase. In the design phase I created the application and data models and decided on a method of application to database interaction. Basically, I began to develop actual methods that served as the base for the rest of the construction of the application. I built a few sample pages that accessed the database utilizing different methods to determine which method best met my needs.

Once the design phase was complete, I moved onto the implementation phase where I began the coding and database creation work. The implementation phase was by far the longest phase in the project (44 days, see Appendix E) and I am happy to say that the phase did not go over its allotted time. In the implementation phase I began to tackle the actual programming that would power the web site. After using non-technical tools like MS Word for so long, Access and FrontPage were welcome sights.

The web programming was done using FrontPage and the database was created with Access. The basic web development and database creation went very well, no doubt

due to the large amount of planning time spent. The most difficult task of the implementation phase was perfecting the interaction of the web and database. I would estimate that over 50% of the implementation phase was spent on getting the correct queries to run against the database.

The database connection and recordset creation methods are, as follows:

```
'1.  Retreive the URL of this page from Server Variables
strURL = Request.ServerVariables("URL")

'2. MapPath of virtual database file path to a physical path.
      strDBPath = "c:\inetpub\wwwroot\fpdb\football.mdb"

'3.  Create an ADO Connection to connect to the database.
      Set cnnSearch = Server.CreateObject("ADODB.Connection")

'4.  This line is for the Access database:
      cnnSearch.Open  "Provider=Microsoft.Jet.OLEDB.4.0;Data  Source="&
strDBPath & ";"
'5. CreateSQL Query
      strSQL = "Select * from current_roster where ID=1;"
'6. Create Recordset Object
      Set rstSearch = Server.CreateObject("ADODB.Recordset")
'7. Open Recordset
      rstSearch.Open  strSQL,  cnnSearch,  adOpenStatic,  adLockReadOnly,
adCmdText
```

The main problems encountered were difficulties with cursor types, recordset characteristics and database locking. Generally, the problems were fixed with the help of 'Beginning Active Server Pages' by Buser, et al (ISBN: 1861003382) and the use of several ASP help web pages: APSworld.com, 4guysfrom rolla.com, etc.

Results and Conclusions

The objective of the project was met by the completion of the following:

- The proposed web application was delivered on time with all planned functionality. As a result of user testing and feedback, additional functionality was also added to improve the user experience.

- The project was approved and signed off upon.

The anticipated deliverable was delivered.

I learned a great deal over the planning, implementation and report writing of this project. First and foremost, I learned how to handle large projects and that careful planning is not only helpful but necessary to complete large projects on time in a way that meets the project's requirements.

Second, I increased my knowledge and level of comfort with ASP and MS Access. The project also helped sharpen my creative programming skills as I was forced to create the entire framework of the project. I had no code examples of how a fantasy football web site was supposed to be designed and coded. Over the course of the project I was also able to hone the skill of re-design. Many of the things that I wanted to do would just simply not work the way I planned. An example of this is viewing player statistics. I began with a detailed plan on how to view the statistics, but in the final version was used because of the implementation of the web to database interaction.

The re-designing was actually a blessing in disguise. I became adept at finding creative ways to move around problems and to design-on-the-fly new methods to finish the project. I'd like to hear a bit more about this.

Also, I realized the value of having end-user feedback. Many of the enhancements that have been performed to the project have been a direct result of feedback from users. Users have a much different perspective on the project and this perspective revealed many shortcomings of the projects as well as giving many ideas on how to improve the project. I think that since I had worked on this project for so many hours I began not to see things how they where but how I thought they were. Having

user input was very valuable because all they could see was the actual pages and not the ideas I had in my brain. Suggestions ranged from 'Reformat this link because I could not tell it was a link' to 'change your access method to improve your page load times'. The user feedback was a valuable source of information.

For future improvements, I would expand the size of the roster to include defensive players. The statistics download from Yahoo provides full statistics on offensive and defensive players so adding defensive players to the league would be no problem. Also, since the project was designed to allow for such enhancements, a major overhaul would not be needed. I think the ideal roster would include: 1 Quarterback, 1 Running Back, 1 Wide Receiver, 1 Defensive Lineman, 1 Linebacker, 1 Defensive Back, 1 Punter and 1 Kicker. This roster design would be very unique and bring new levels of planning and theory to the week-to-week management of each participant's roster.

I would also improve small things like the layout of the pages to make them easier to use and work with. Unfortunately, my programming skills far exceed my ability to work with the layout of objects on a page and make them look presentable. What I am getting at here is that the layout of the pages is not an ideal layout. Generally, I think the pages could look much better with a few small adjustments. The basic navigation structure is easy to follow and no advanced technologies were used that would bring about browser problems users, such as Flash or Java applets.

I would also incorporate the use of a chat room or message board so the participants can communicate with one another. Allowing the participants to communicate would increase the fun and competitiveness of the game and would allow participants to offer help to inexperienced participants.

References

Buser, D., Kauffman, J., Llibre, J. T., Francis, B., Sussman, D., Ullman, C., & Duckett, J. Beginning Active Server Pages 3.0. (1999). Wrox Press Ltd. Birmingham, UK.

Elmasri, R. Navathe, S. B. (2000). Fundamentals of Database Systems (3rd Ed.). Addison Wesley, Reading MA.

Internet Information Services Features. (Online). Accessed on: 4/9/2002. Available: http://www.microsoft.com/windows2000/server/evaluation/features/web.asp

Microsoft OLE DB White Paper. (Online). Accessed on: 4/9/2002. Available: http://www.microsoft.com/data/oledb/prodinfo/wpapers/wpapers.htm

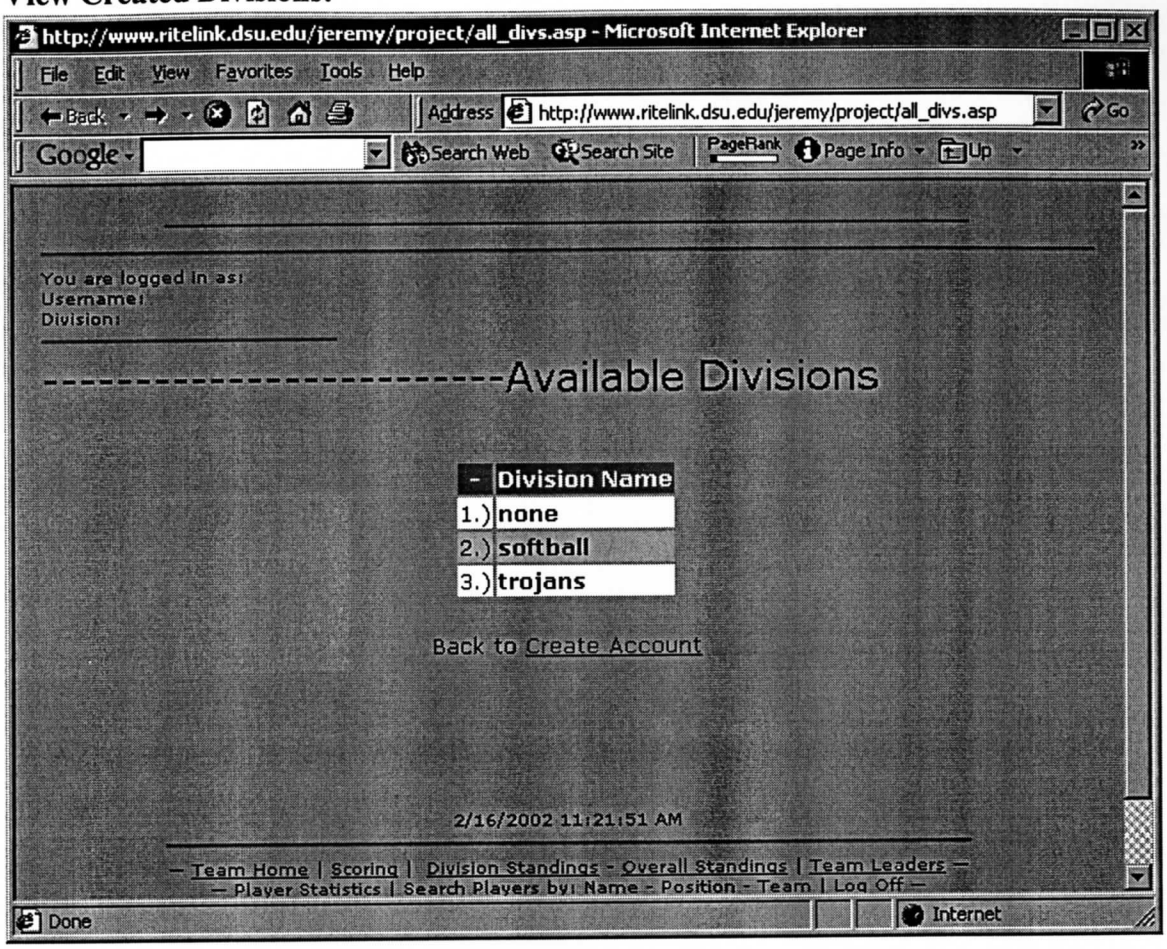
Appendix A Application Screen Shots:

- Application Login (home.htm) p.A-1
- Create Account (new_user.asp) p.A-2
- View Created Divisions (all_divs.asp)p.A-3
- Team or Home (main.asp)p.A-4
- Player Search by Name (db_search.asp) p.A-5
- Player Search by Position (db_serach_pos.asp) p.A-6
- Player Search by Team (db_Search_team.asp) p.A-7
- View Player Statistics (full_stats_find.asp) p.A-8
- View NFL Leaders (nfl_leaders.asp) p.A-9
- View Team Leaders (team_leaders.asp) p.A-10
- Administrator Home Page p. A-11
- Administrator Edit Users p.A-14
- Administrator Delete Users p.A-15
- Administrator Update Statistics p.A-12 & A-13
- View Overall Standings p.A-16
- View Weekly NFL Leader p. A-17
- View Weekly Player Statistics p. A-18
- View Opponent's Roster p. A-18
- View Division Standings p.A-20
- View Weekly User Leaders p. A-21 & A-22

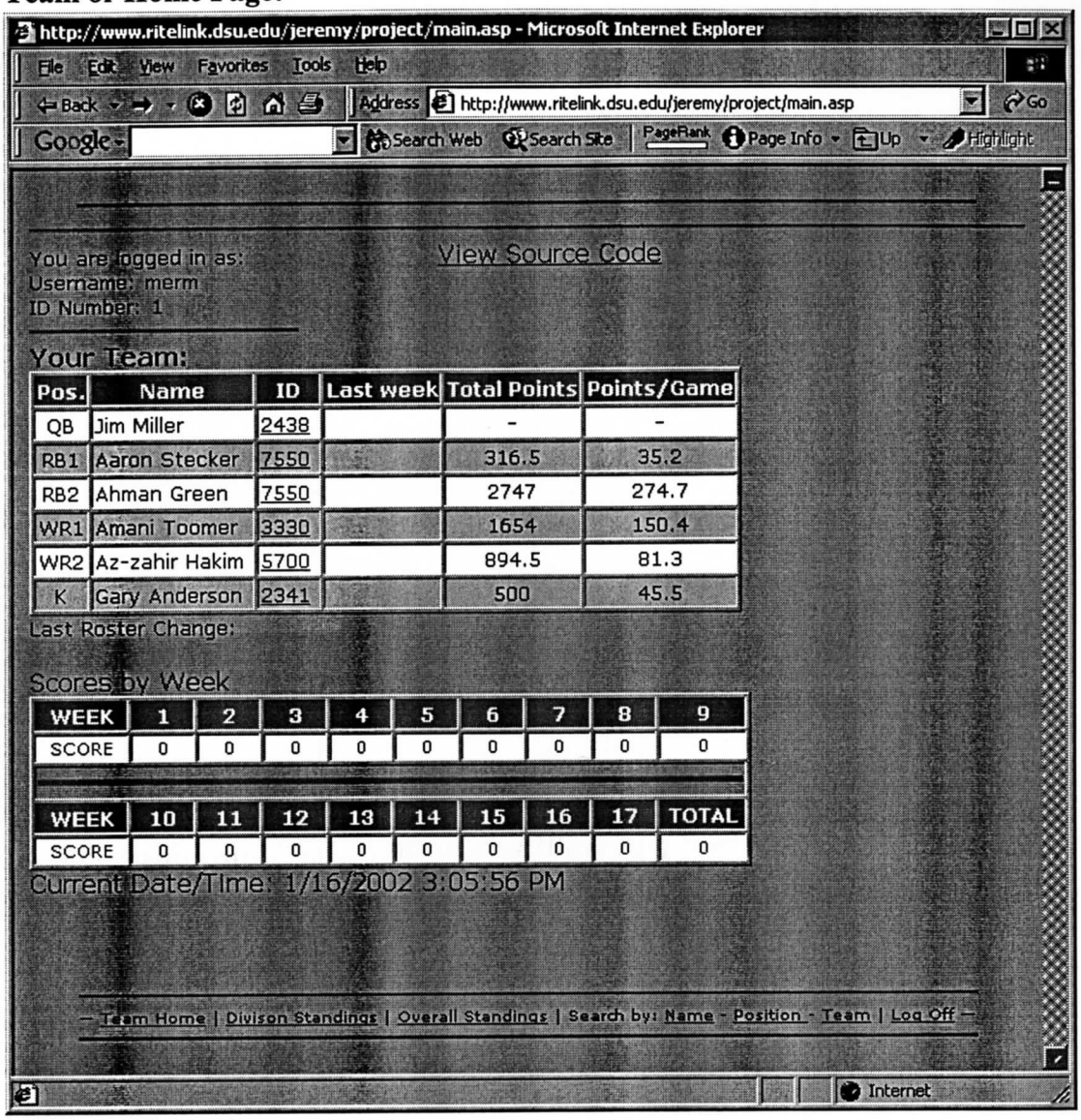
Application Login:

Create Account Screen:

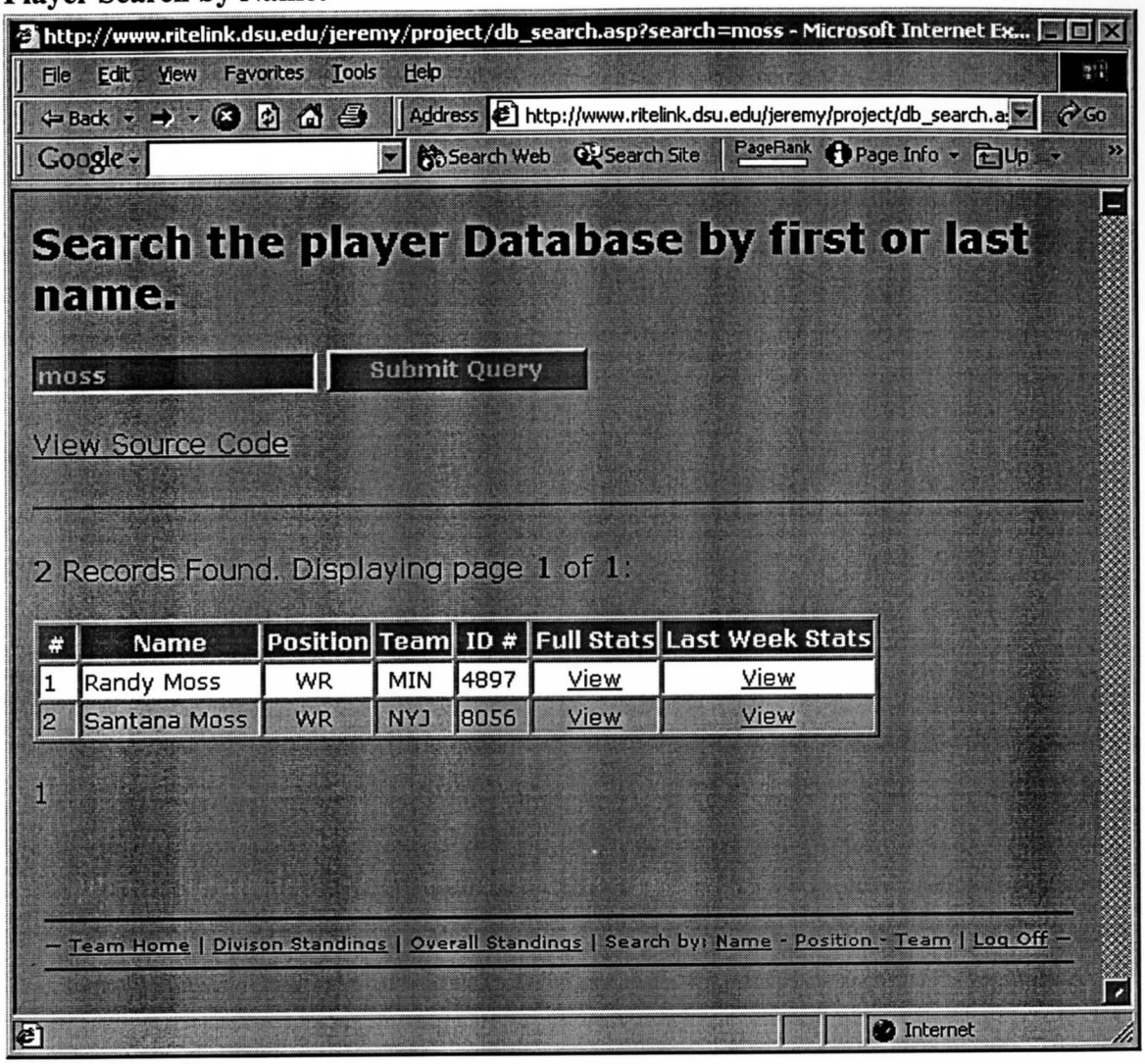
View Created Divisions:



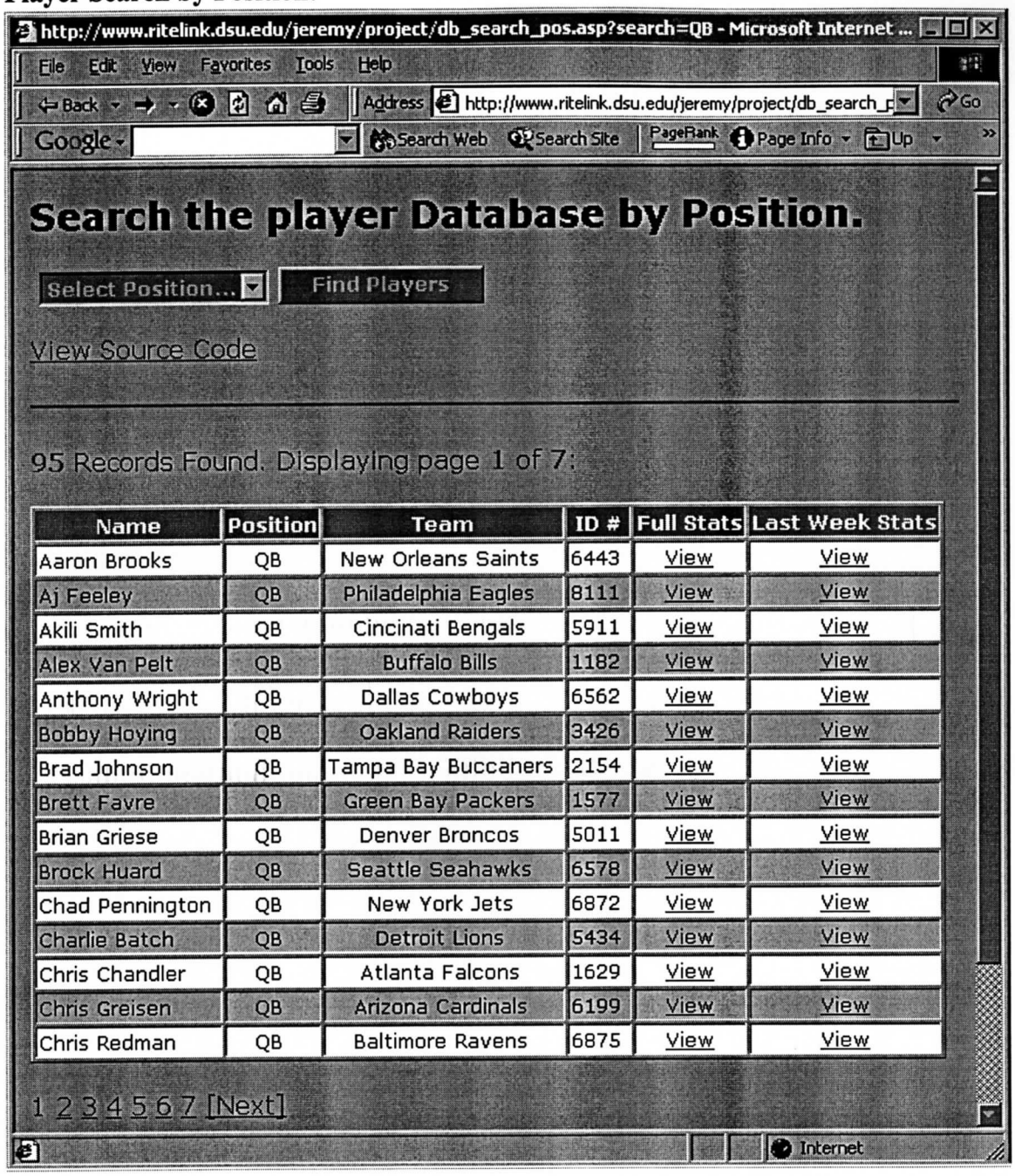
Team or Home Page:



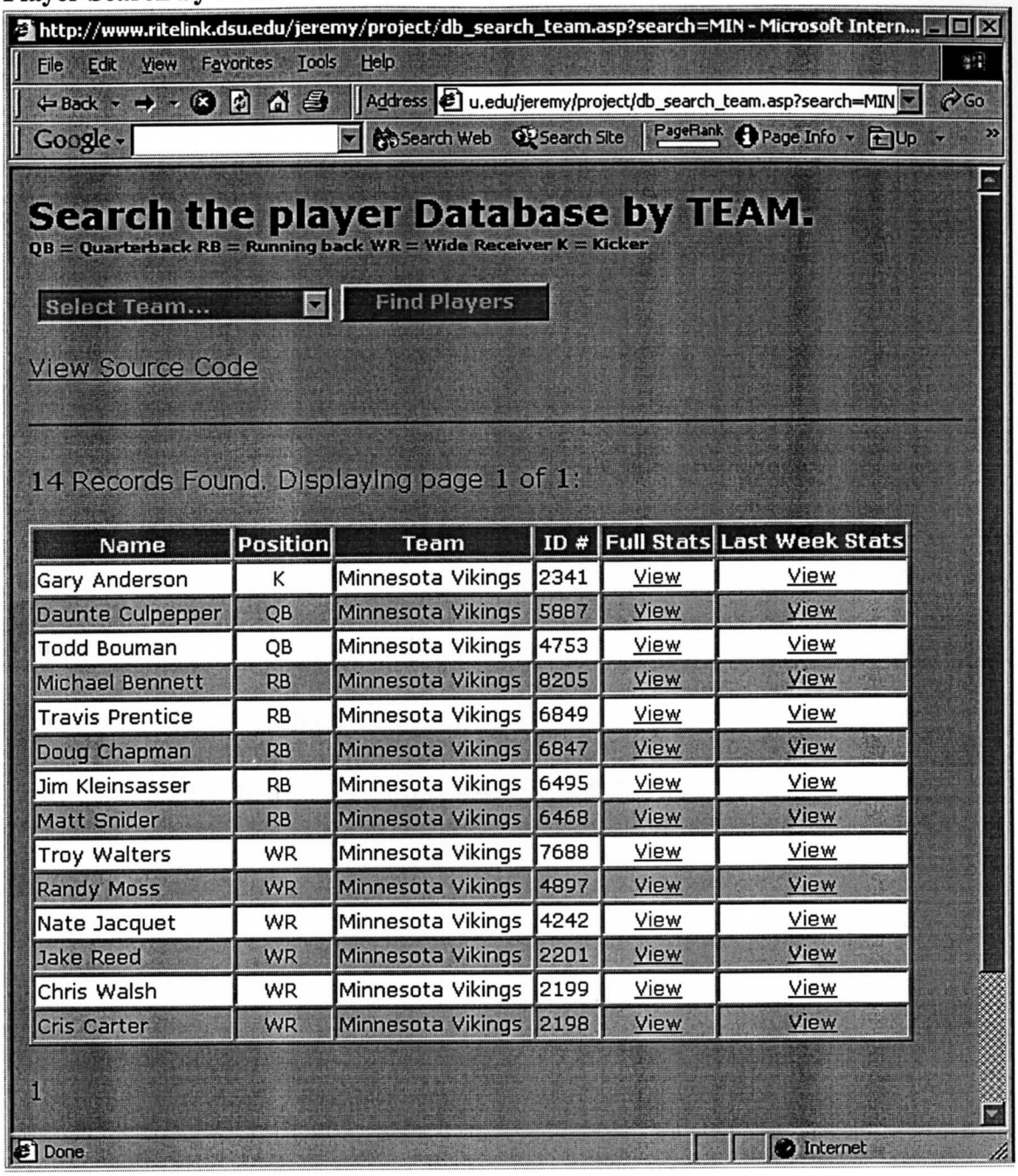
Player Search by Name:



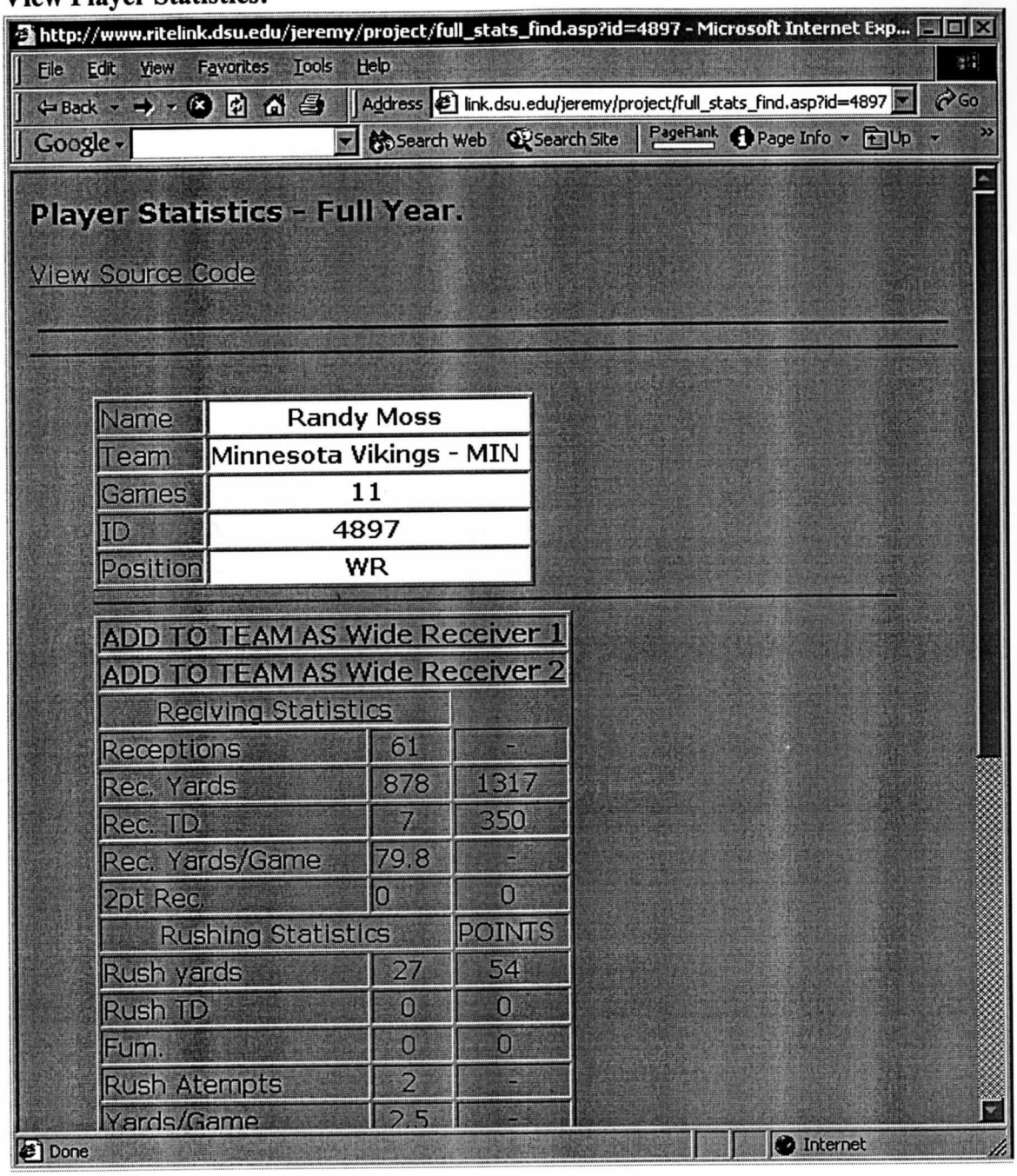
Player Search by Position:



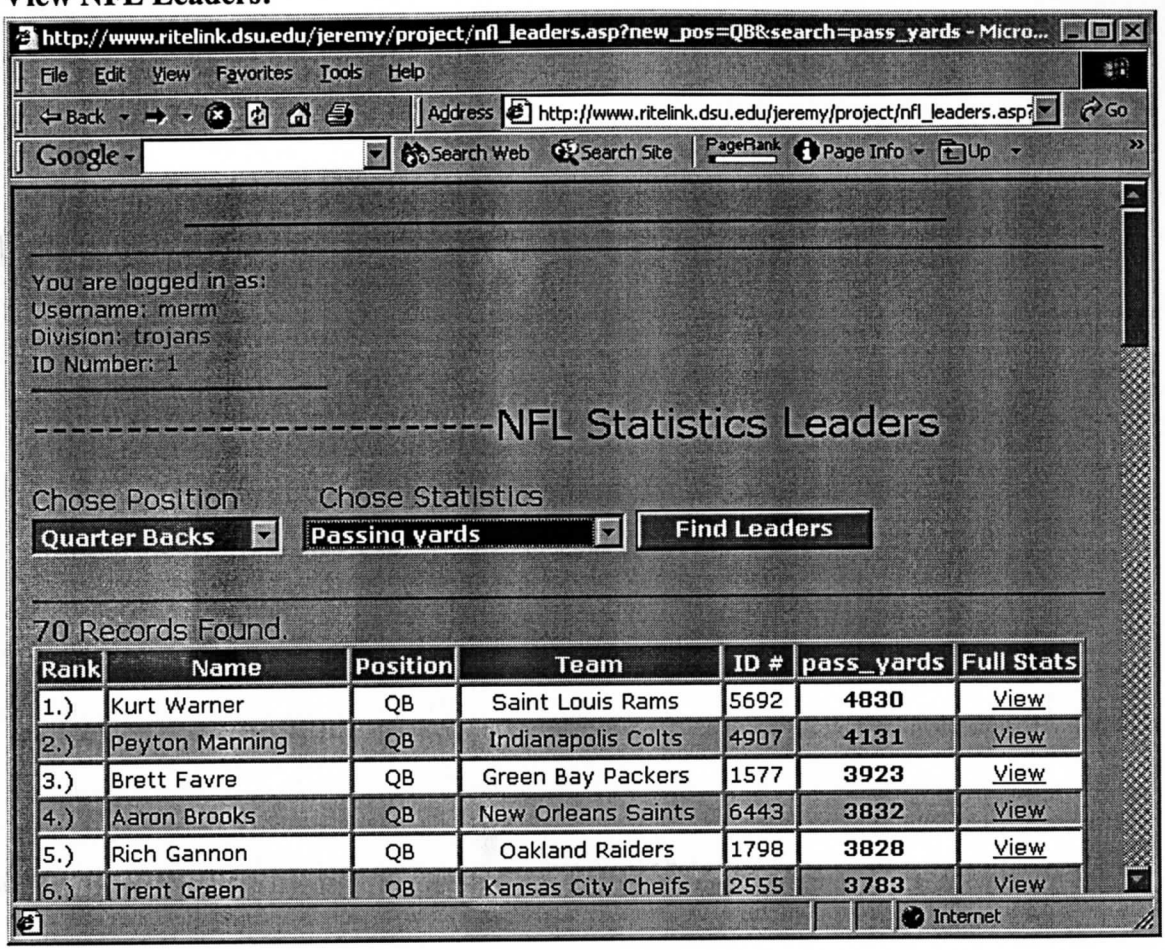
Player Search by Team:



View Player Statistics:

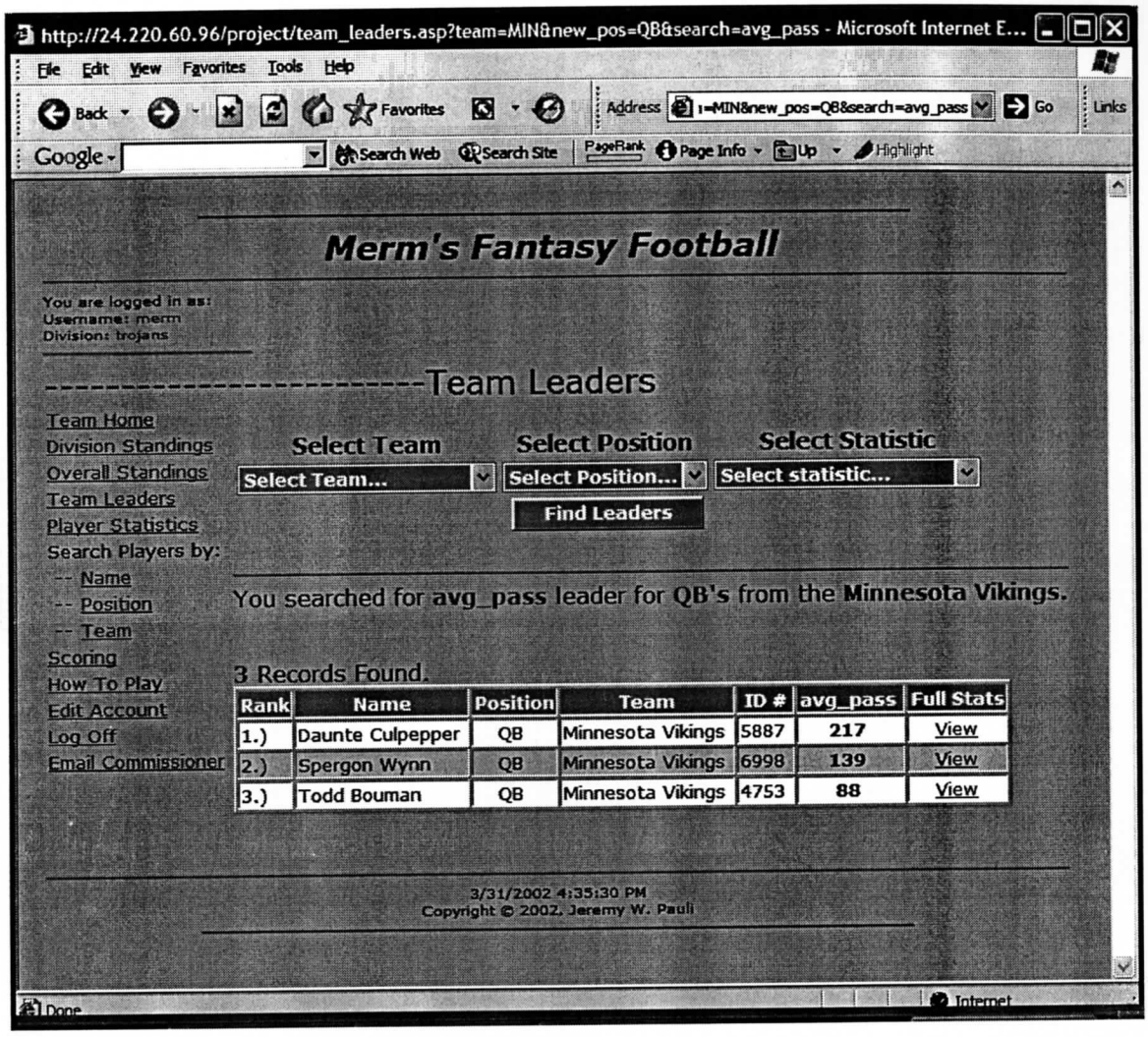


View NFL Leaders:

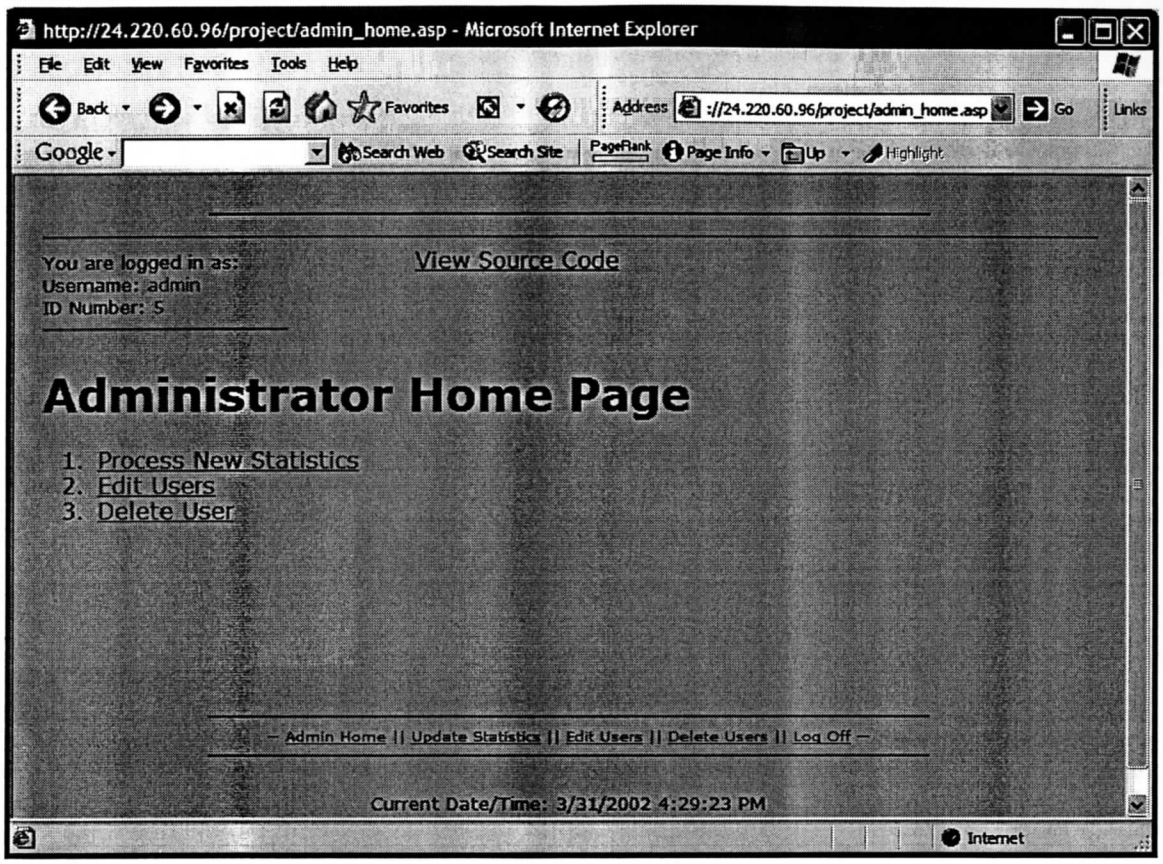


Rank	Name	Position	Team	ID #	pass_yards	Full Stats
1.)	Kurt Warner	QB	Saint Louis Rams	5692	**4830**	View
2.)	Peyton Manning	QB	Indianapolis Colts	4907	**4131**	View
3.)	Brett Favre	QB	Green Bay Packers	1577	**3923**	View
4.)	Aaron Brooks	QB	New Orleans Saints	6443	**3832**	View
5.)	Rich Gannon	QB	Oakland Raiders	1798	**3828**	View
6.)	Trent Green	QB	Kansas City Cheifs	2555	**3783**	View

View Team Leaders

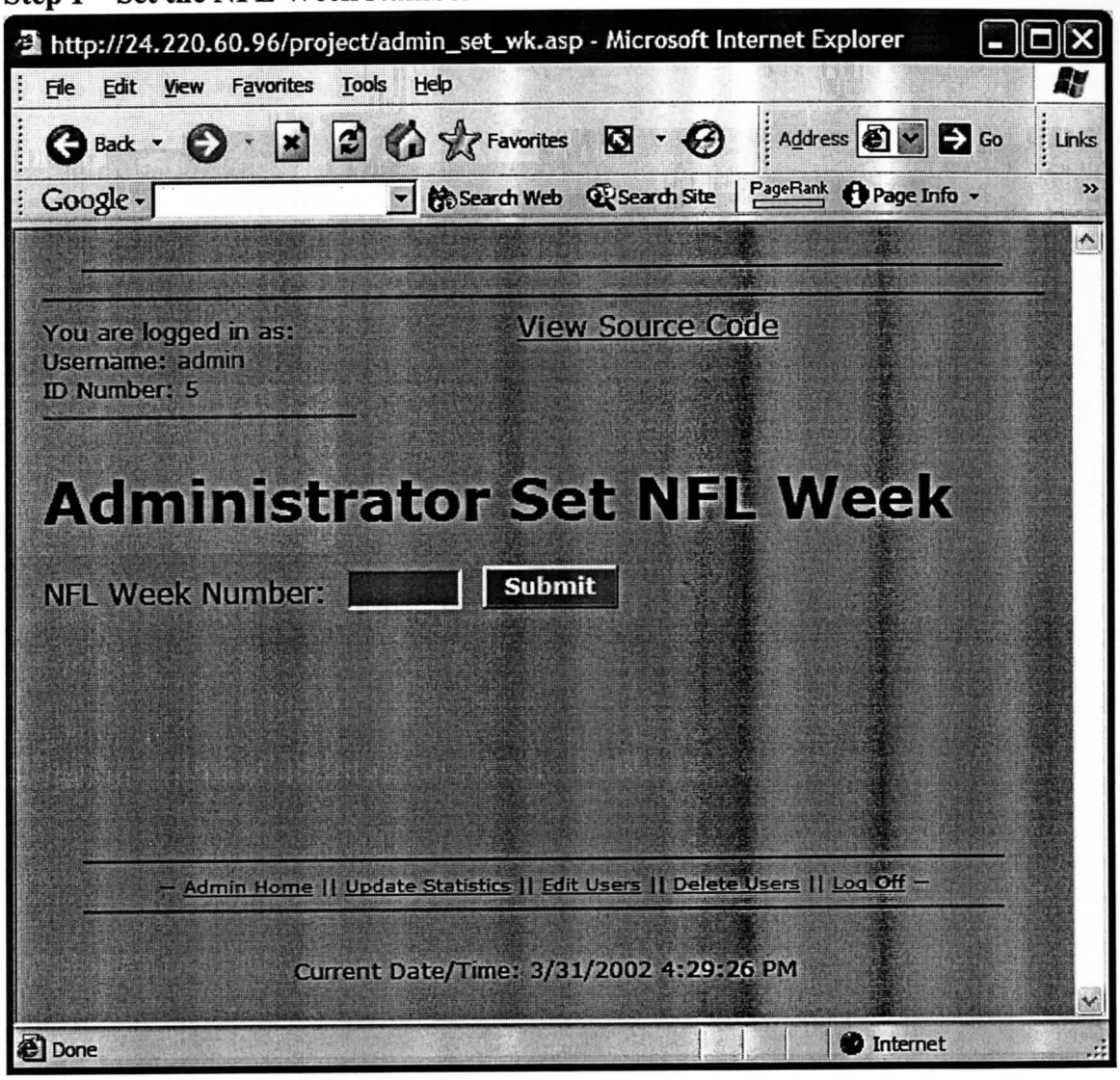


Administrator Home:

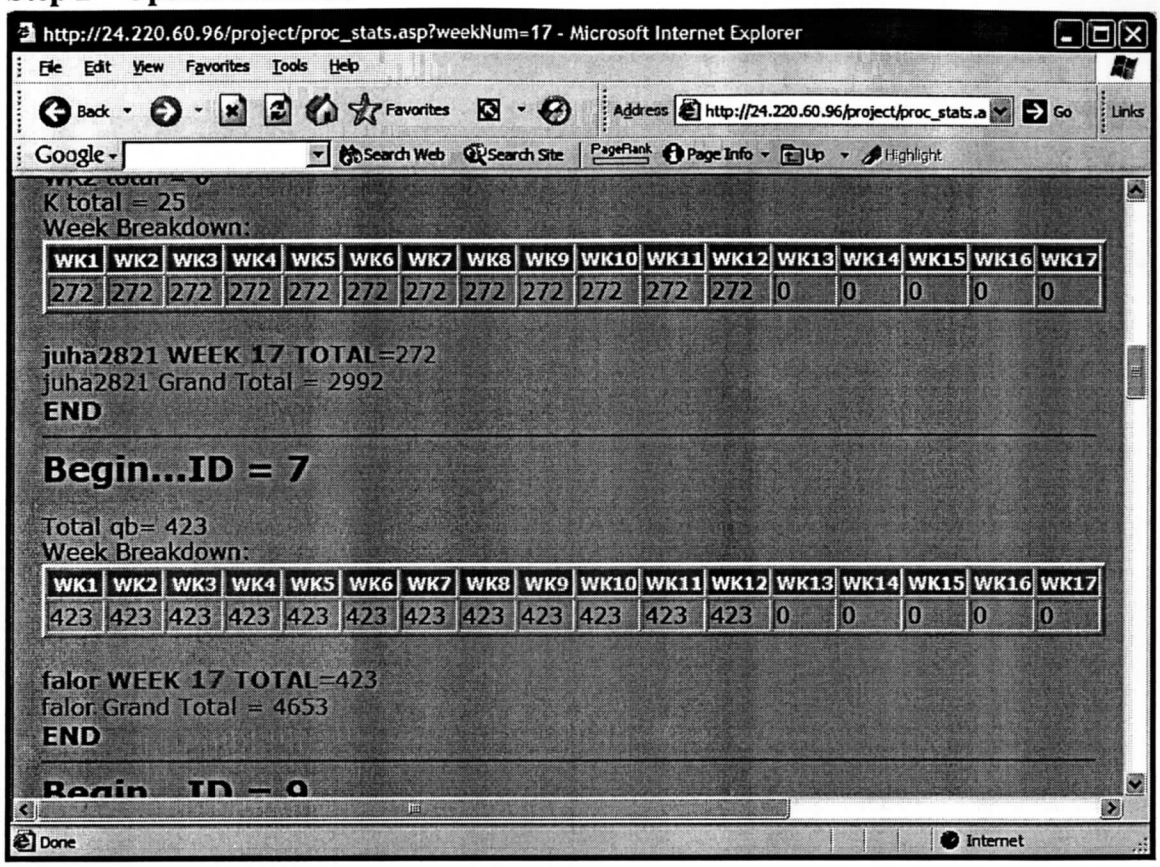


Administrator Update Statistics:

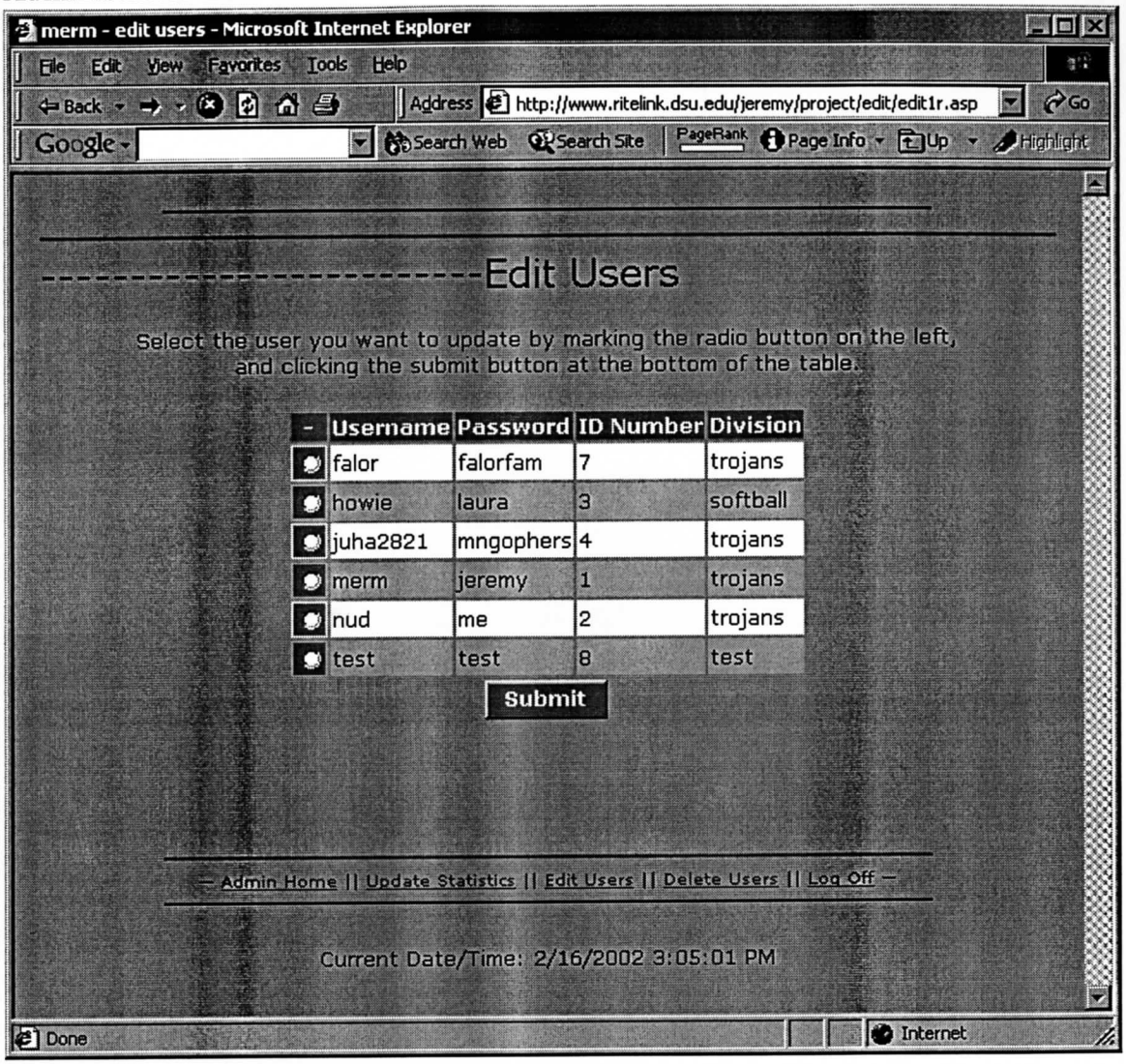
Step 1 – Set the NFL Week Number



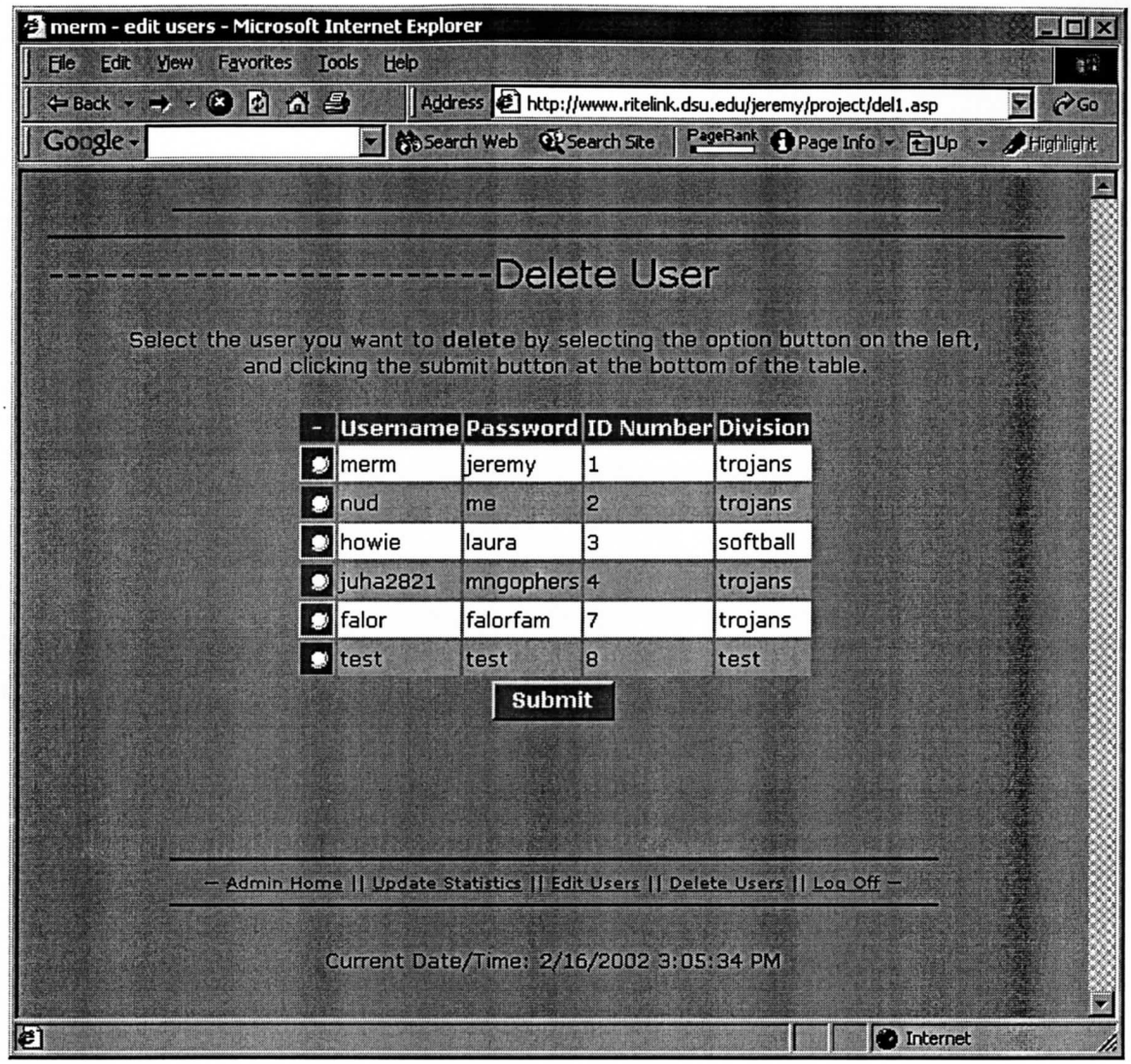
Step 2 – Update Statistics



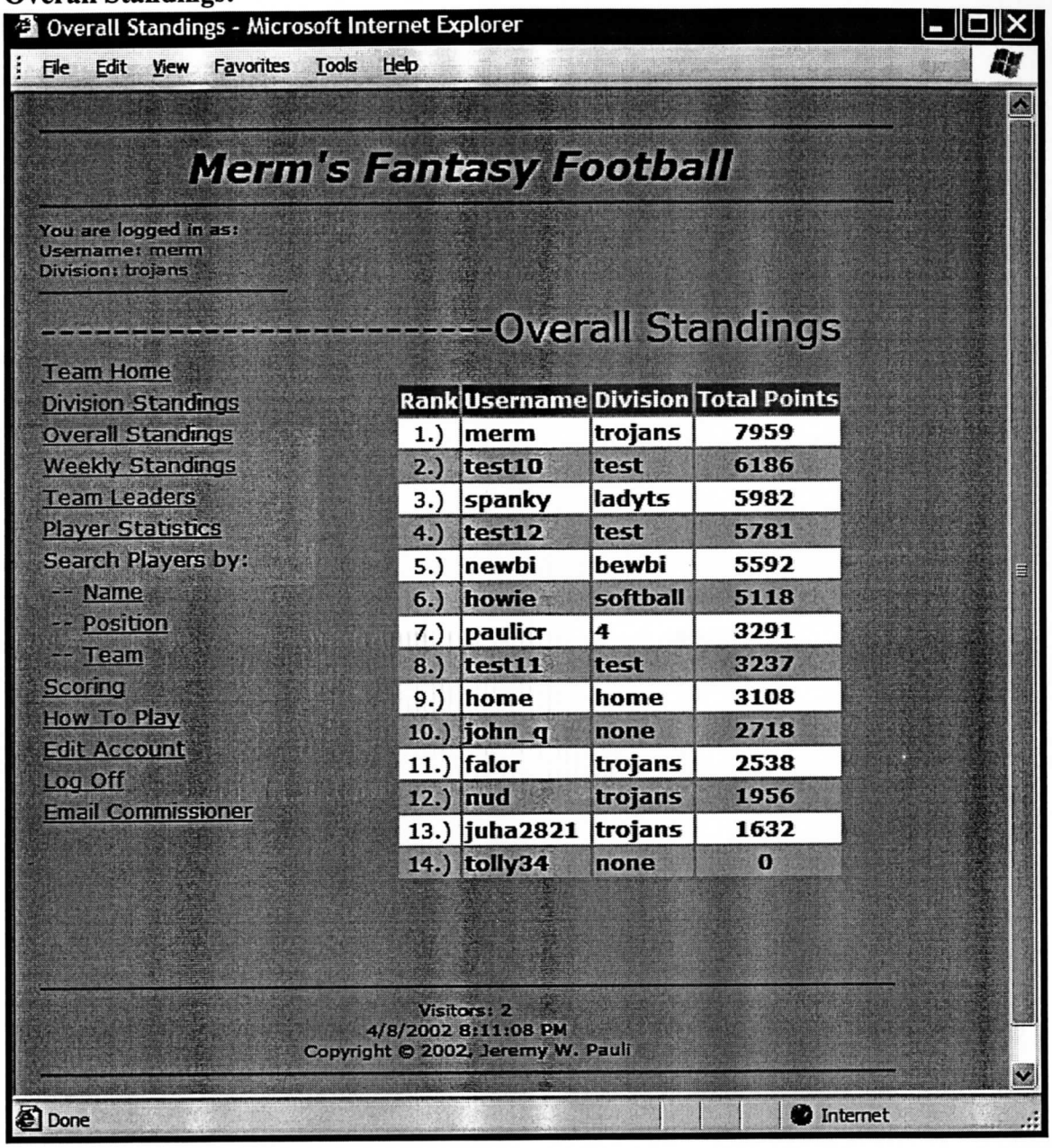
Administrator Edit Users:



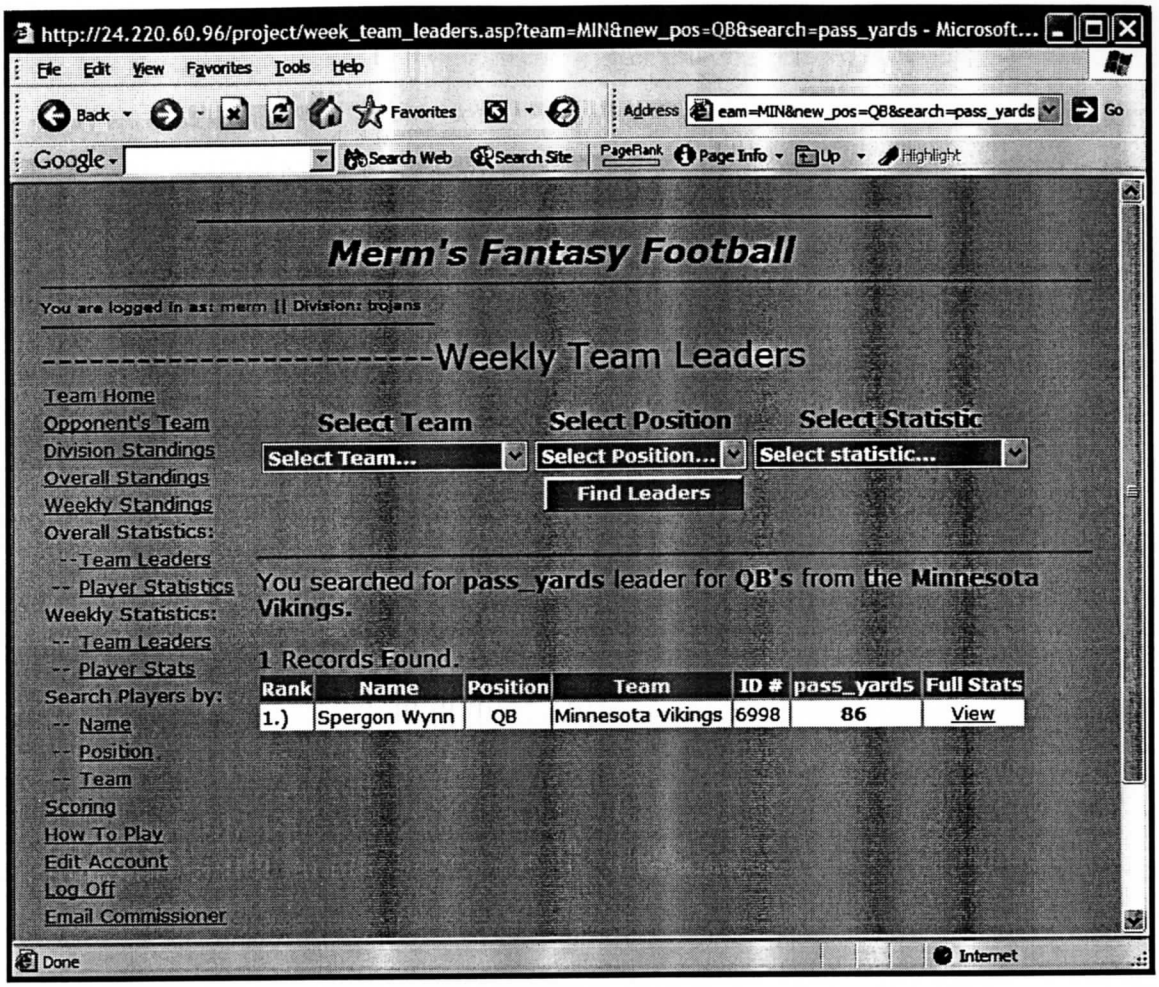
Administrator Delete Users:



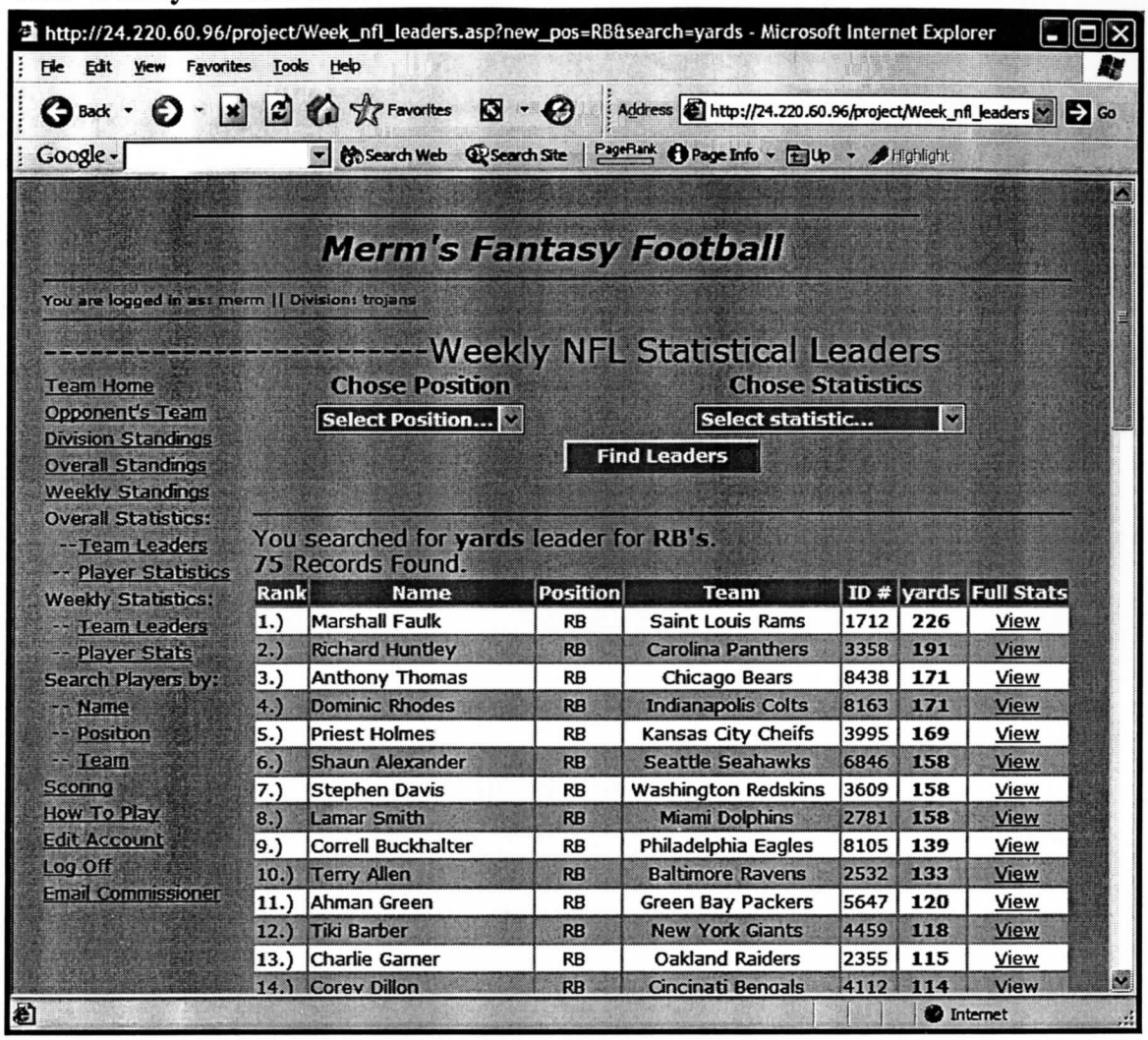
Overall Standings:



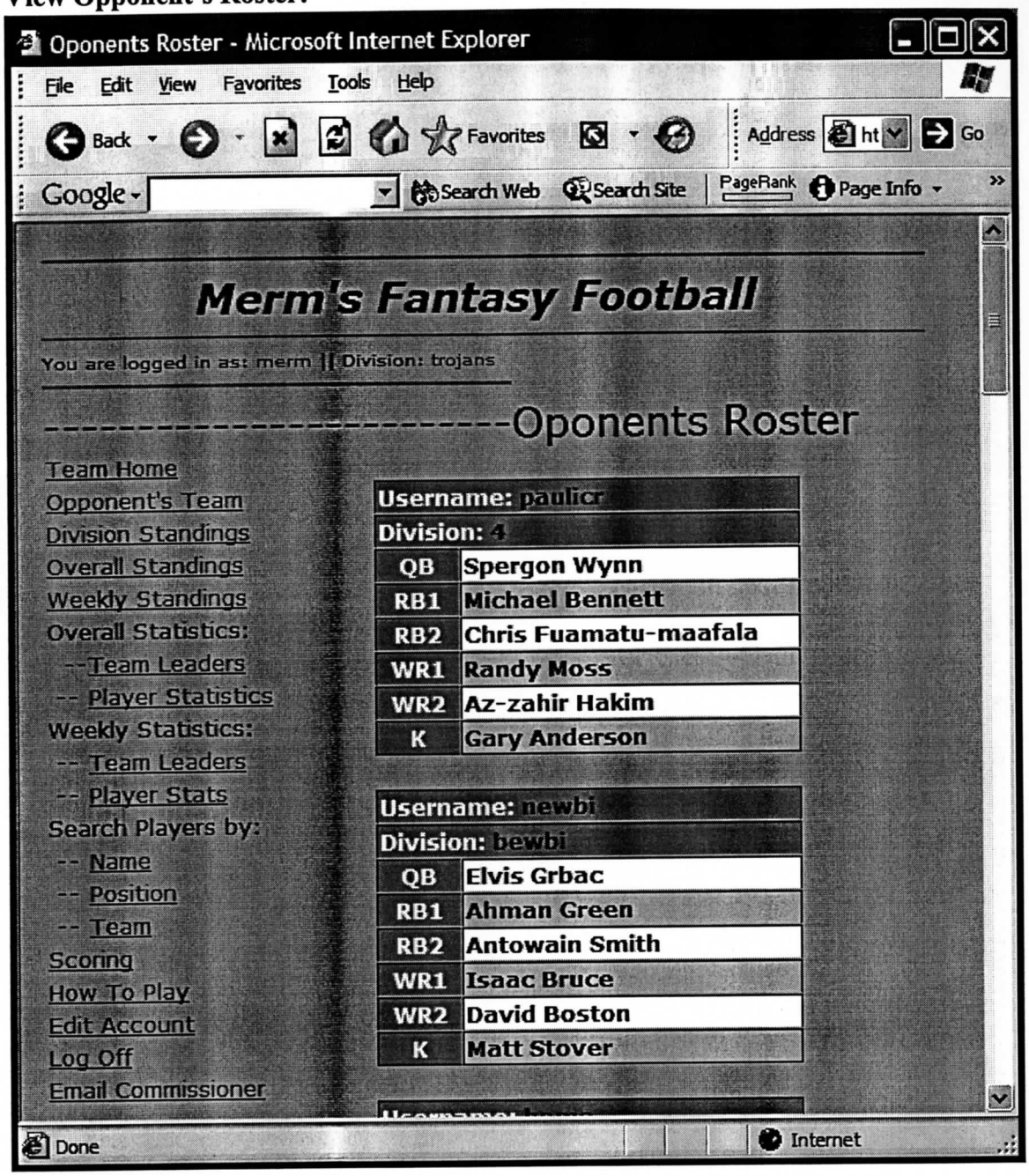
View Weekly Team Leaders:



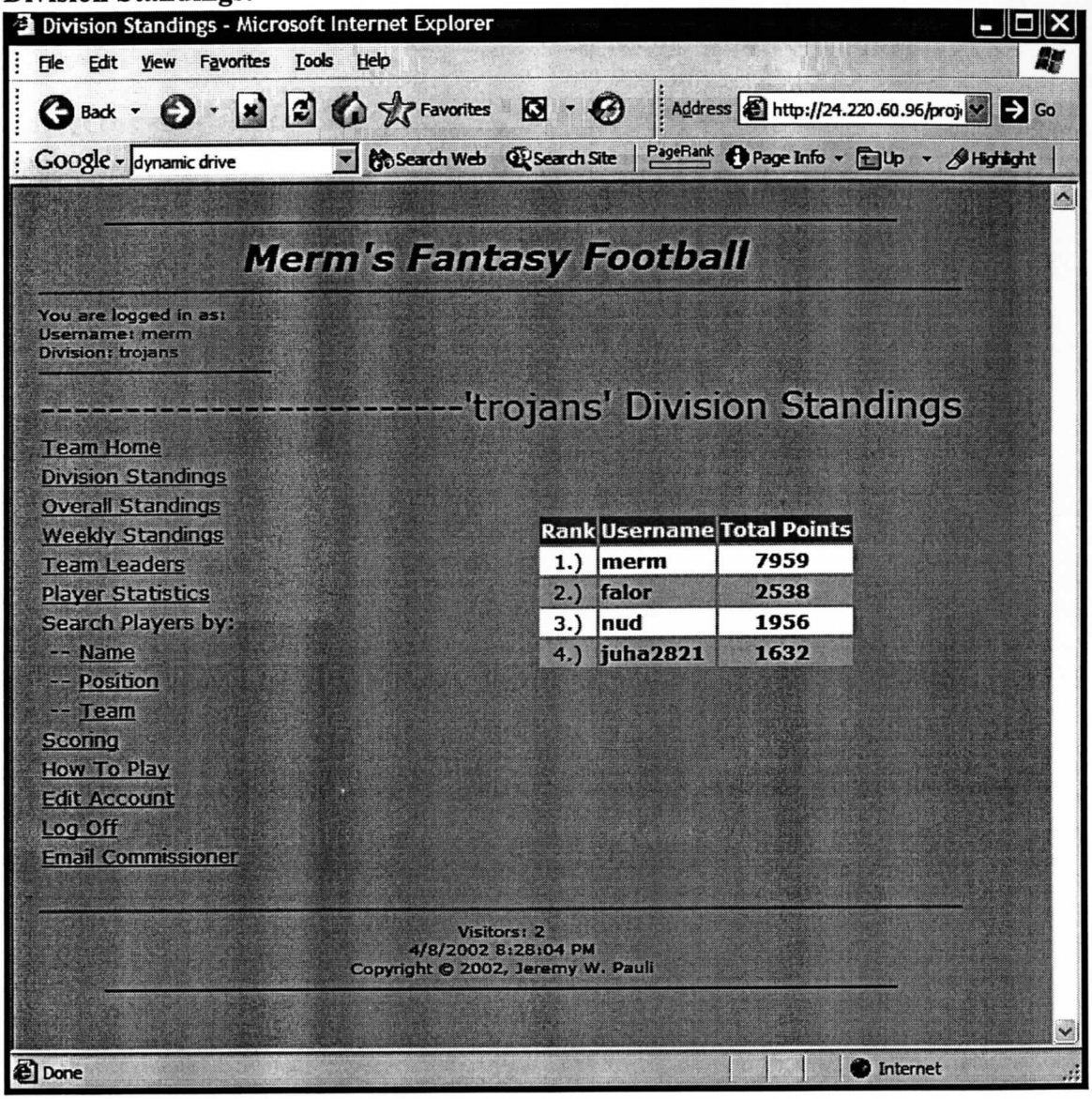
View Weekly NFL Leaders:



View Opponent's Roster:

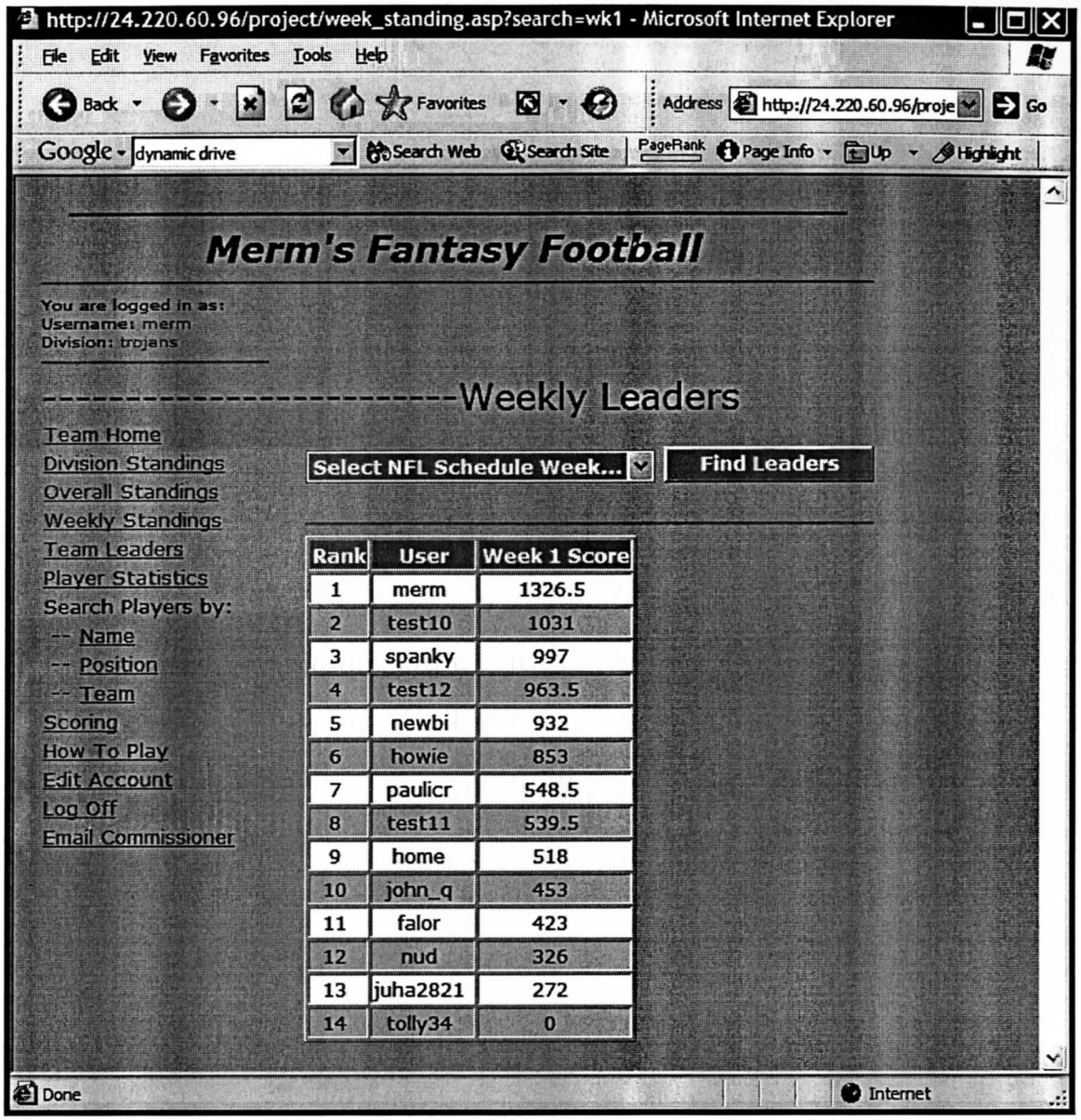


Division Standings:



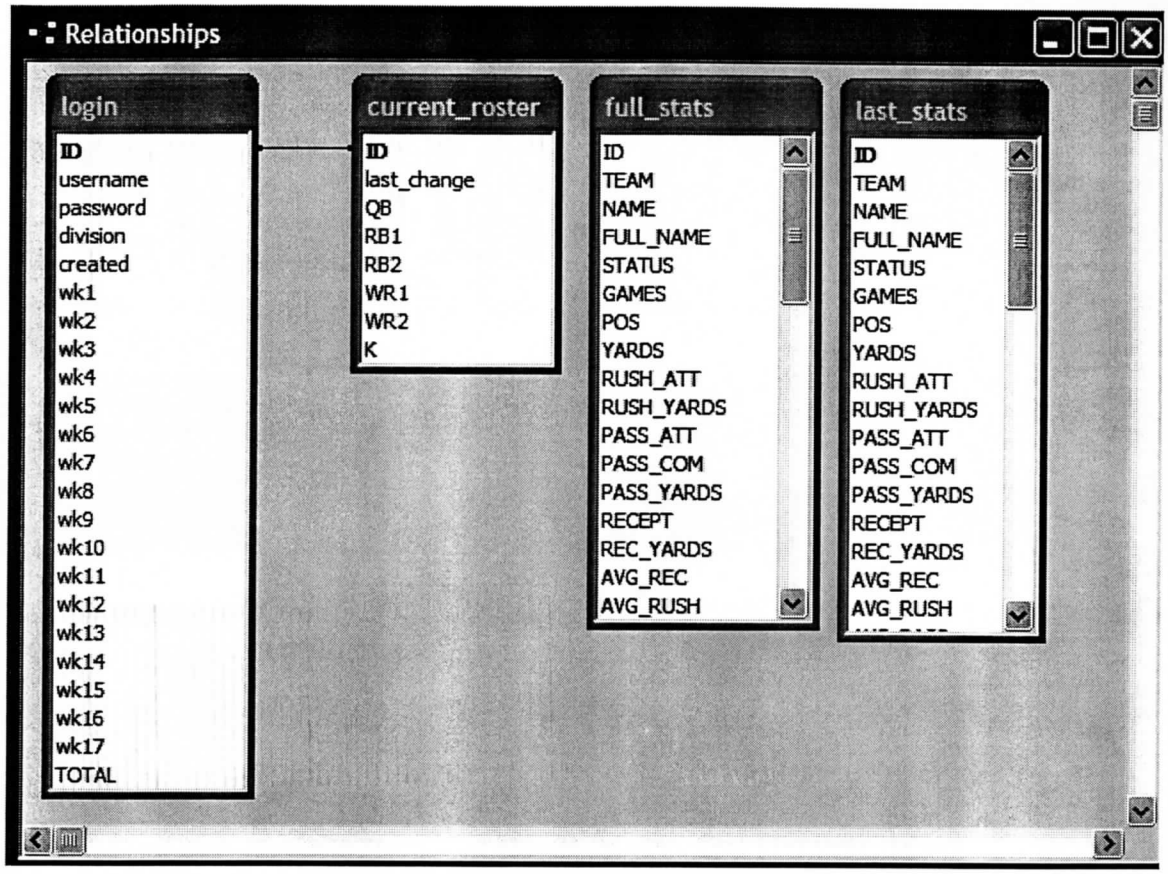
Weekly Standings:
Step 1 – Select Week Number and pres 'Find Leaders'

Result:


http://24.220.60.96/project/week_standing.asp?search=wk1 - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Favorites Address http://24.220.60.96/proje Go
Google dynamic drive Search Web Search Site PageRank Page Info Up Highlight
Merm's Fantasy Football
You are logged in as:
Username: merm
Division: trojans
Weekly Leaders
Team Home
Division Standings
Overall Standings
Weekly Standings
Team Leaders
Player Statistics
Search Players by:
-- Name
-- Position
-- Team
Scoring
How To Play
Edit Account
Log Off
Email Commissioner
Select NFL Schedule Week...
Find Leaders
Rank User Week 1 Score
1 merm 1326.5
2 test10 1031
3 spanky 997
4 test12 963.5
5 newbi 932
6 howie 853
7 paulicr 548.5
8 test11 539.5
9 home 518
10 john_q 453
11 falor 423
12 nud 326
13 juha2821 272
14 tolly34 0
Done
Internet

Appendix B: Database Design, Schema, Downloading Statistics/importing into Access and Optimization

DATABASE DESIGN:



DATABASE TABLE NAMES AND DESCRIPTIONS:

TABLE NAME	DESCRIPTION
Last_stats	Table that holds the most recent weekly stats for the NFL players – imported from Yahoo.com. X indicates the week number from the NFL schedule (1-17).
full_stats	Total stats for NFL players updated to the most recent week – imported from Yahoo.com.
Current_roster	Holds the most current user rosters for each user.
Login	Holds the Login information for each user as well as the points scored each week and total points.

DATABASE TABLES AND FIELDS:

TABLE: *login*

ID	Username	Password	division	Week1-17*	total

*One field for each week of the NFL season

TABLE: *full_stats* (imported from Yahoo.com)

Name	Player_ID	Team	position	total	games	Individual stats* (54 fields total)

*Example: rush_yards, pass_yards, etc.

TABLE: *last_stats* (where x is the week number; table imported from Yahoo.com)

Name	Player_ID	Team	position	total	Individual stats* (46 fields total)

*Example: rush_yards, pass_yards, etc.

TABLE: *current_roster*

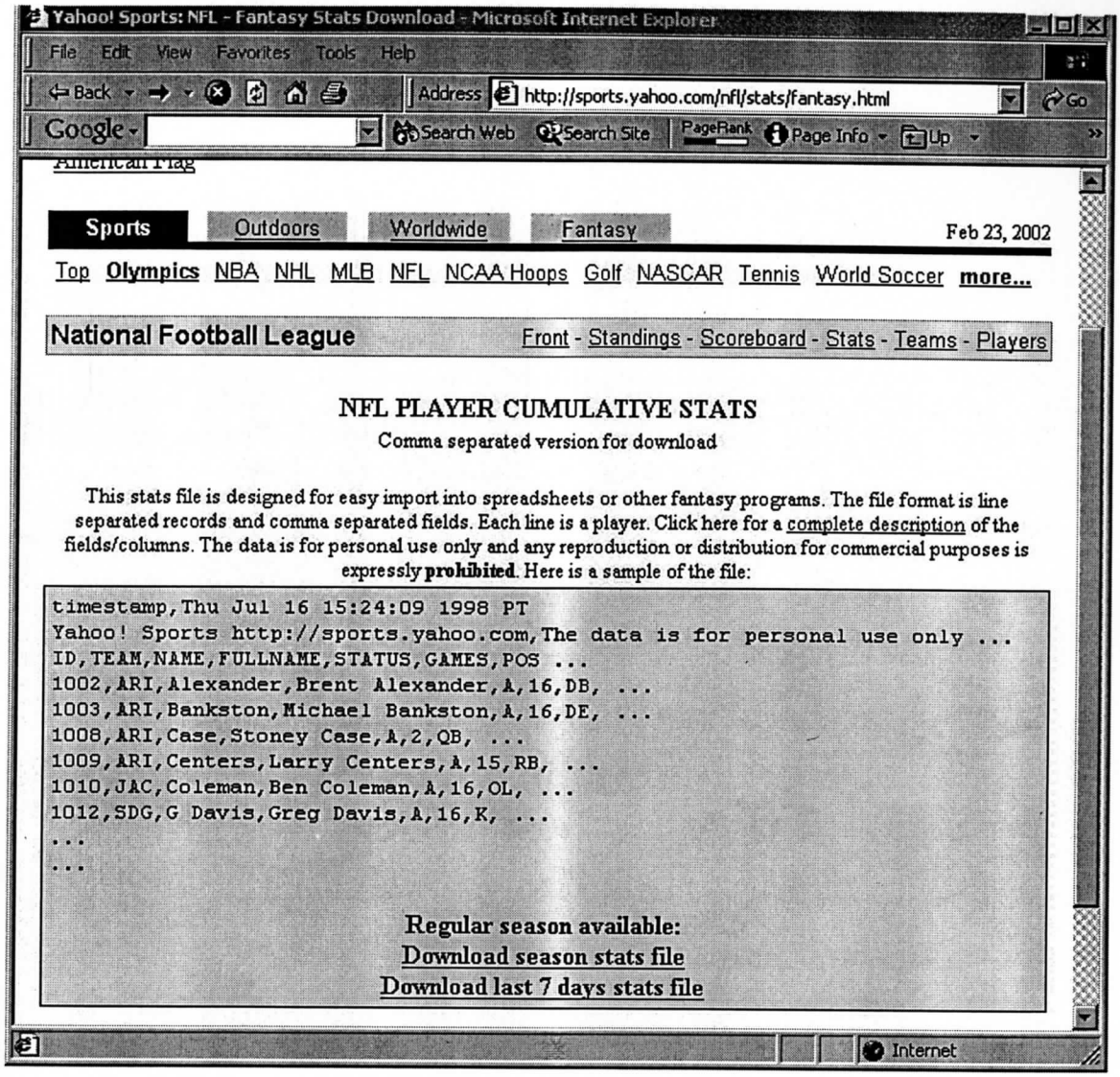
ID	QB	RB1	RB2	WR1	WR2	K

Downloading and importing the statistics from Yahoo.com

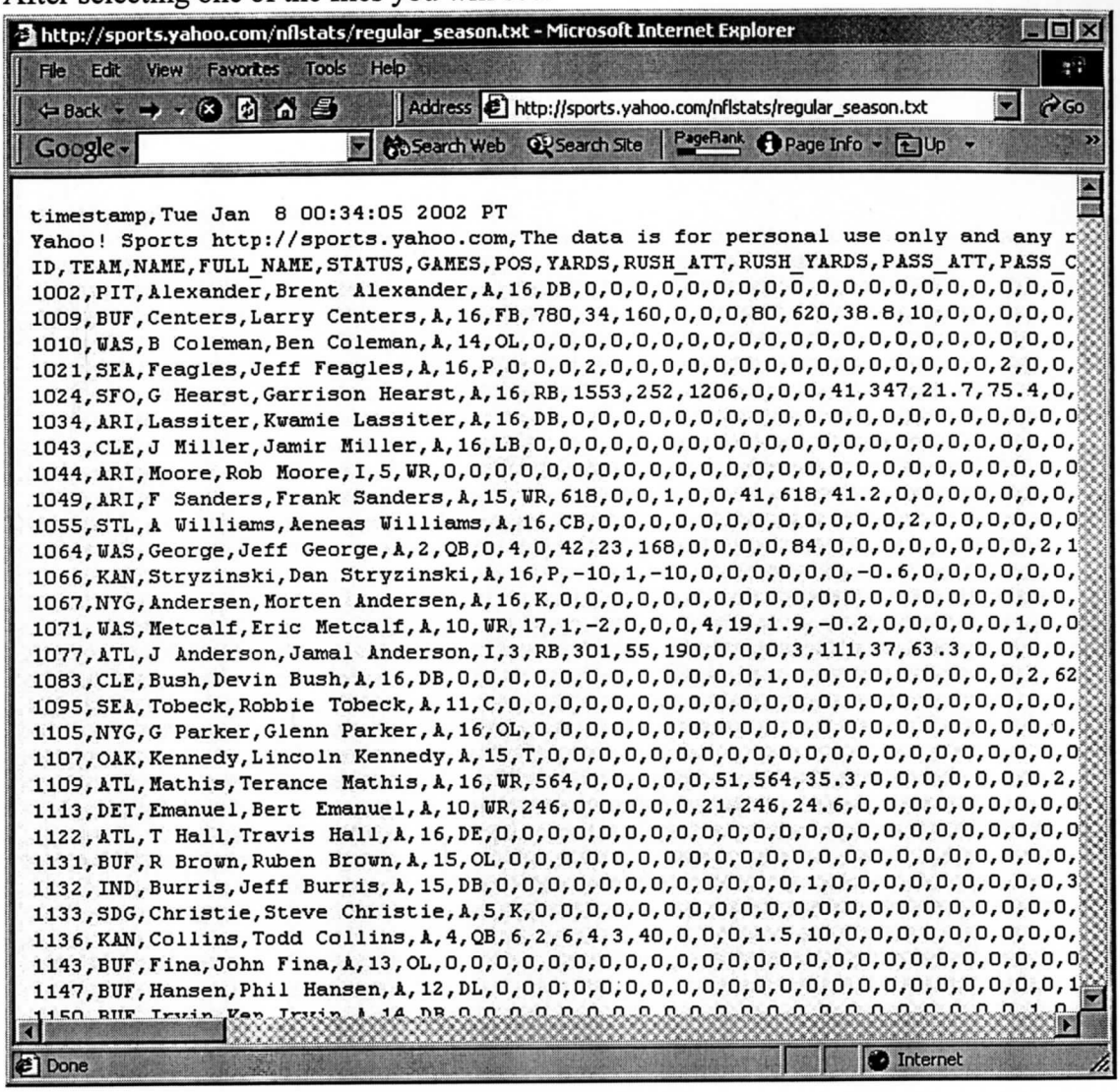
The NFL fantast football statistics can be freely downloaded at: http://sports.yahoo.com/nfl/stats/fantasy.html

Two files are required each week: one for the statistics for each week and one for the cumulative statistics. The files are downloaded as comma-delimited text files. Once downloaded both files must be imported into MS Access. The process for downloading and importing the files is as follows:

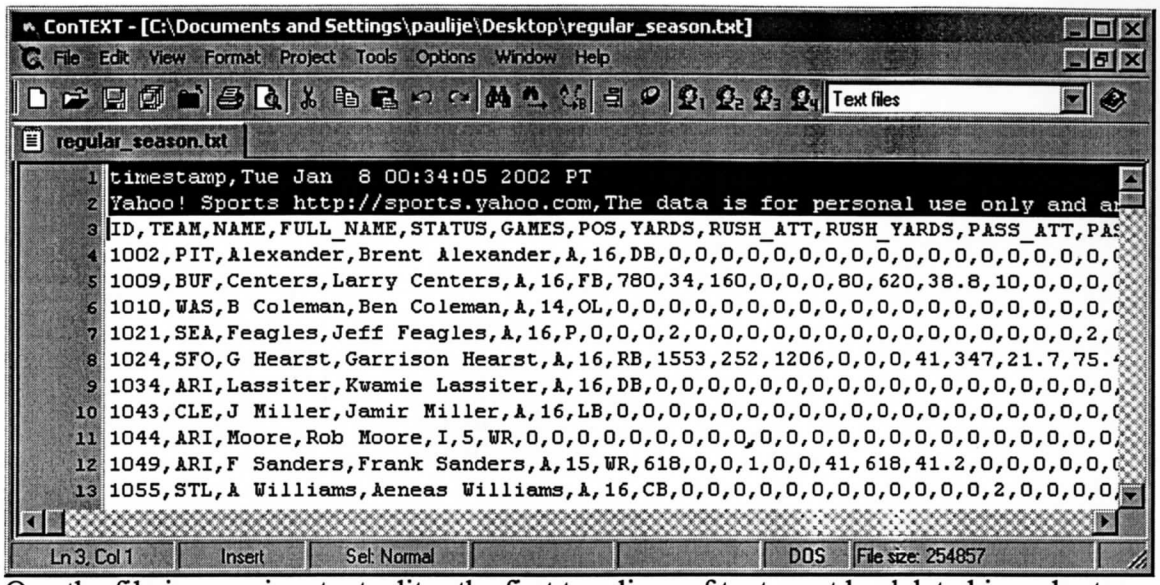
Go to http://sports.yahoo.com/nfl/stats/fantasy.html and select either 'Dowload Season Stats file' or 'Download last 7 days stats file'.


Yahoo! Sports: NFL - Fantasy Stats Download - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Address http://sports.yahoo.com/nfl/stats/fantasy.html
Go
Google
Search Web
Search Site
PageRank
Page Info
Up
American Flag
Sports
Outdoors
Worldwide
Fantasy
Feb 23, 2002
Top Olympics NBA NHL MLB NFL NCAA Hoops Golf NASCAR Tennis World Soccer more...
National Football League
Front - Standings - Scoreboard - Stats - Teams - Players
NFL PLAYER CUMULATIVE STATS
Comma separated version for download
This stats file is designed for easy import into spreadsheets or other fantasy programs. The file format is line separated records and comma separated fields. Each line is a player. Click here for a complete description of the fields/columns. The data is for personal use only and any reproduction or distribution for commercial purposes is expressly prohibited. Here is a sample of the file:
timestamp,Thu Jul 16 15:24:09 1998 PT
Yahoo! Sports http://sports.yahoo.com,The data is for personal use only ...
ID,TEAM,NAME,FULLNAME,STATUS,GAMES,POS ...
1002,ARI,Alexander,Brent Alexander,A,16,DB, ...
1003,ARI,Bankston,Michael Bankston,A,16,DE, ...
1008,ARI,Case,Stoney Case,A,2,QB, ...
1009,ARI,Centers,Larry Centers,A,15,RB, ...
1010,JAC,Coleman,Ben Coleman,A,16,OL, ...
1012,SDG,G Davis,Greg Davis,A,16,K, ...
...
...
Regular season available:
Download season stats file
Download last 7 days stats file
Internet

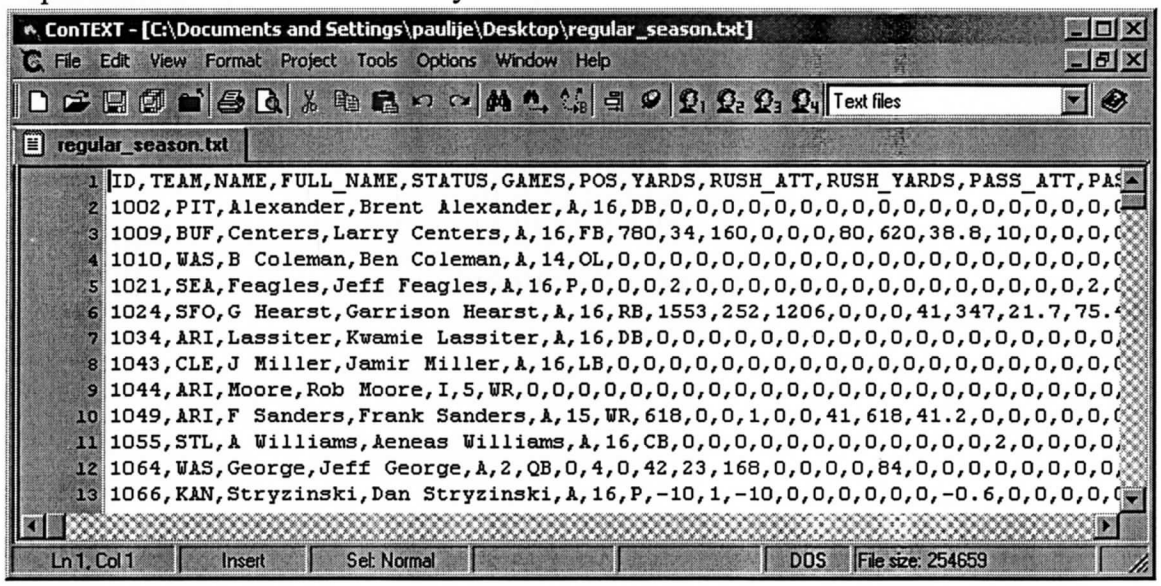
After selecting one of the files you will see:



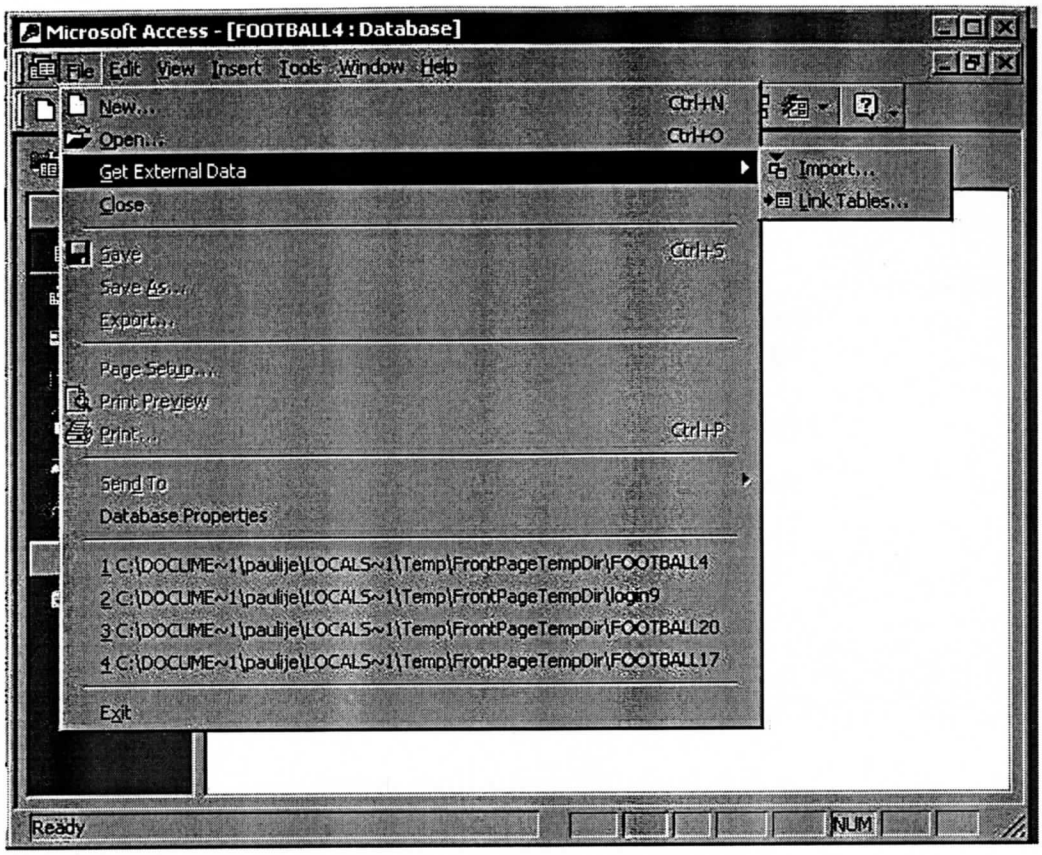
Next select 'Save As' from the 'File' Menu and save the file to disk. Next we will make a slight modification to the file to eliminate a few lines of unneeded text. Open the file in your favorite text editor.



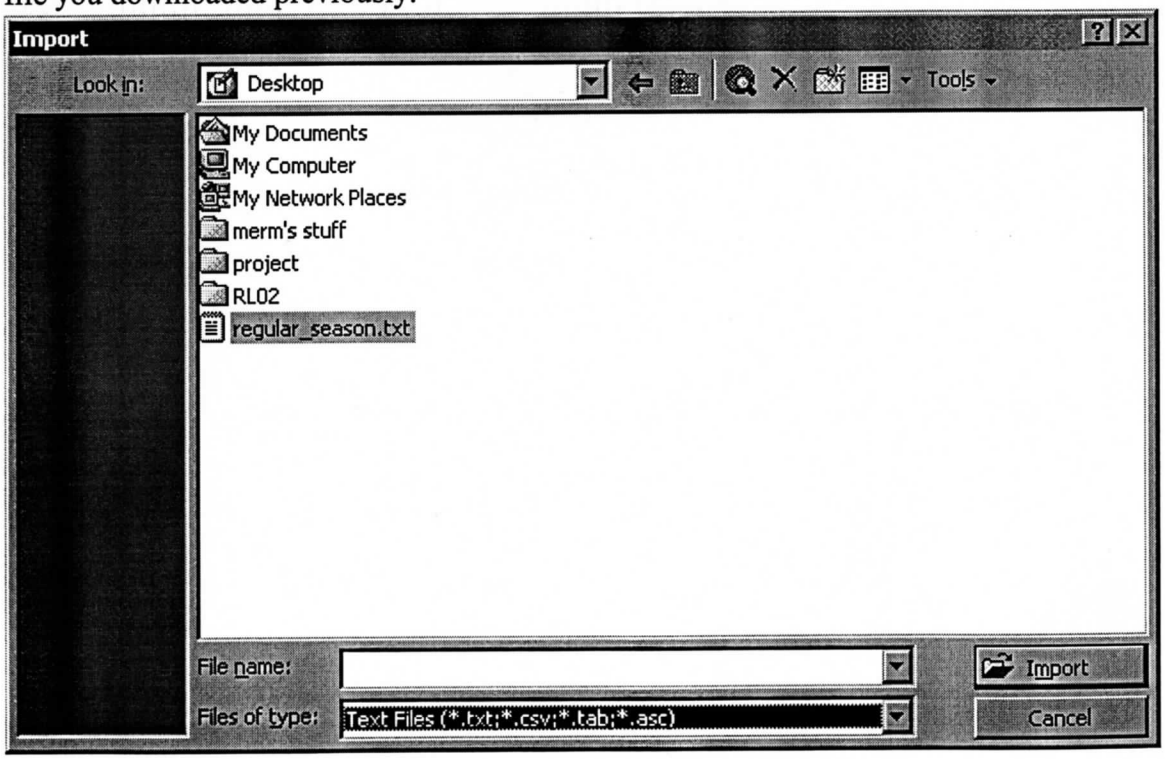
One the file is open in a text editor the first two lines of text must be deleted in order to import the file into Access correctly. Once done the file should look like:



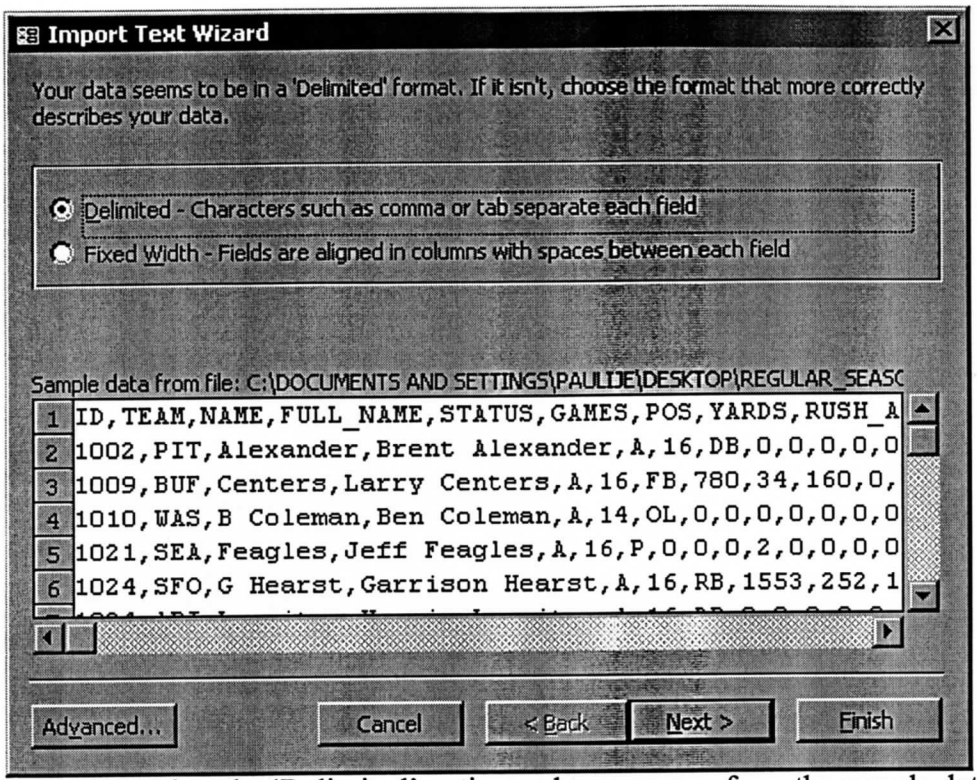
Next open Access and select the 'File' Menu, then the 'Get External Data' Option and then finally the 'Import' option.



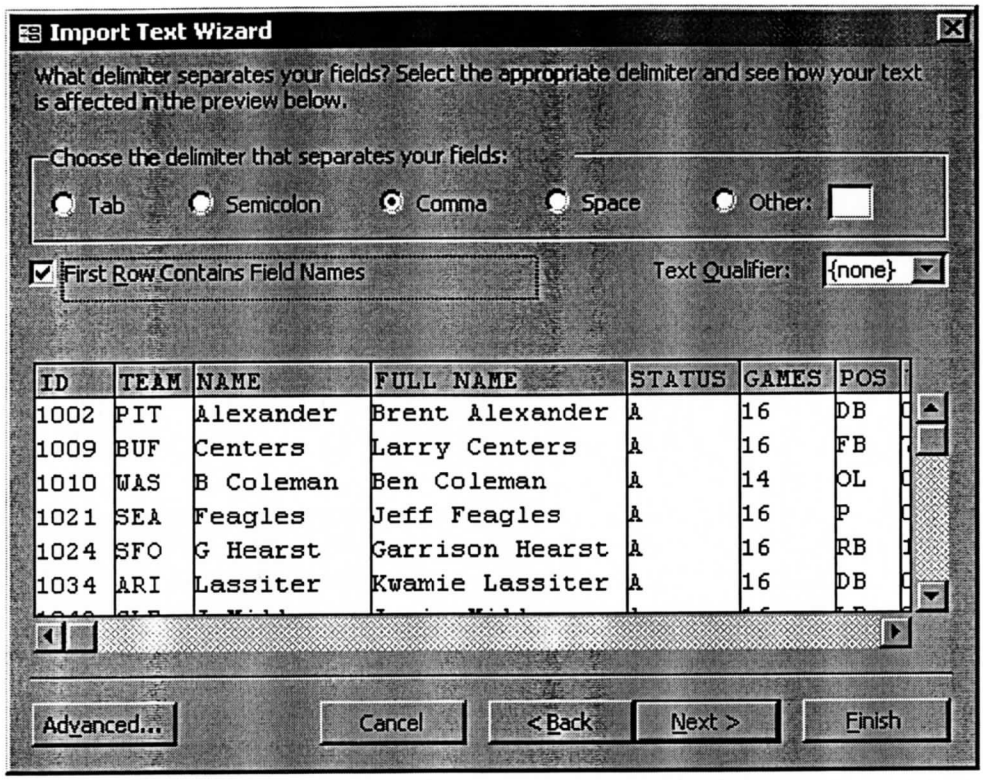
Once selected you will be prompted to select the file you wish to import. Find the text file you downloaded previously.



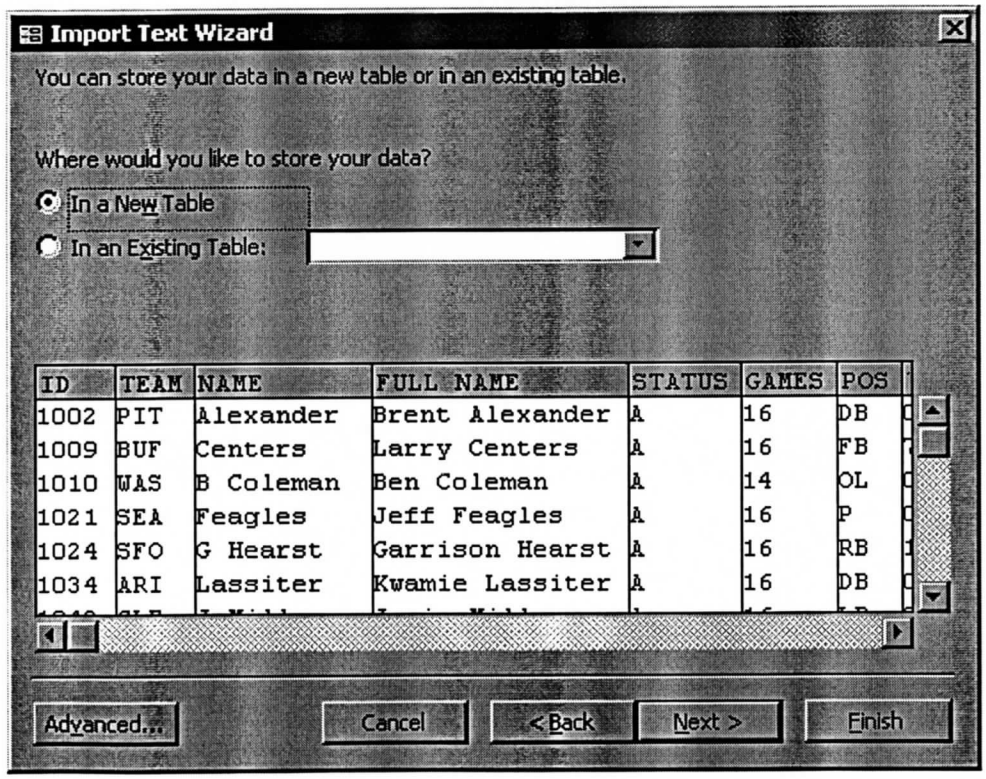
After you select your text file you will see the 'Import Text Wizard'



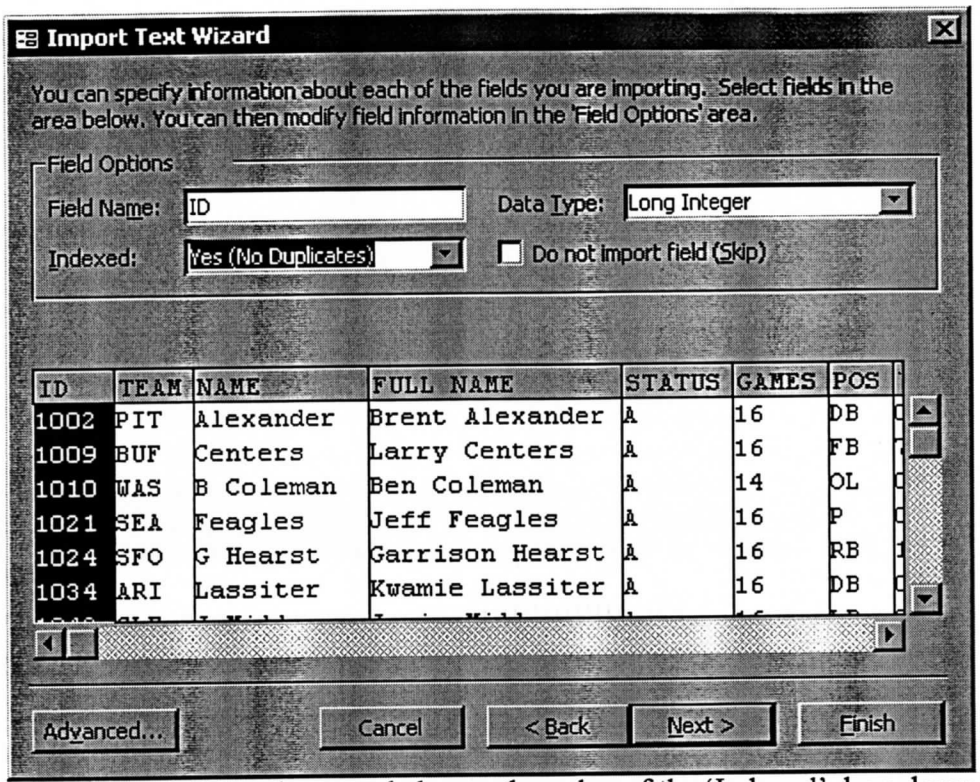
Be sure to select the 'Delimited' option and you can see from the sample data screen that our information is being interpreted correctly so far. Press the 'Next' button to continue.



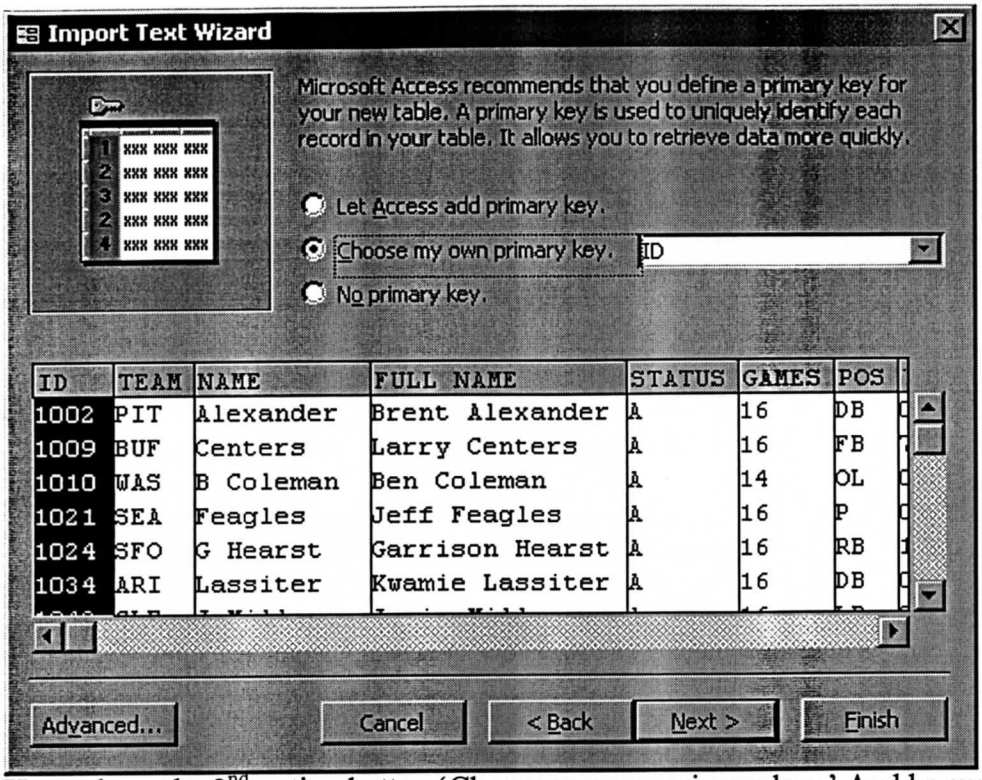
Here we need to be sure that we check the 'First Row Contains Field Names' check box and be sure that the 'comma' option is also selected. Once this is done our data should now look like a typical database table. Press the 'Next' button to continue.



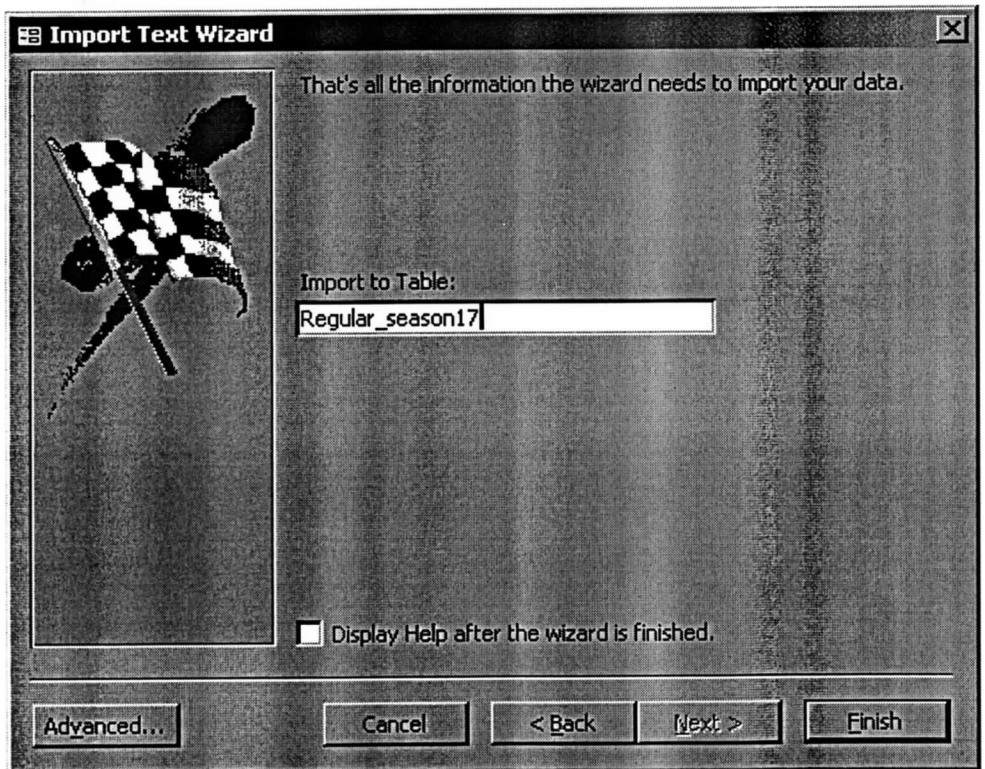
The only option to worry about here is to select if the imported data will go into an existing table or into a new table. I always chose to put the data into a new table to avoid any possible data problems. Press the 'Next' button to continue.



Next select the ID column and change the value of the 'Indexed' drop down box to read 'Yes (No Dulicates)' so we can use the ID field as the primary key for the table. Press the 'Next' button to continue.

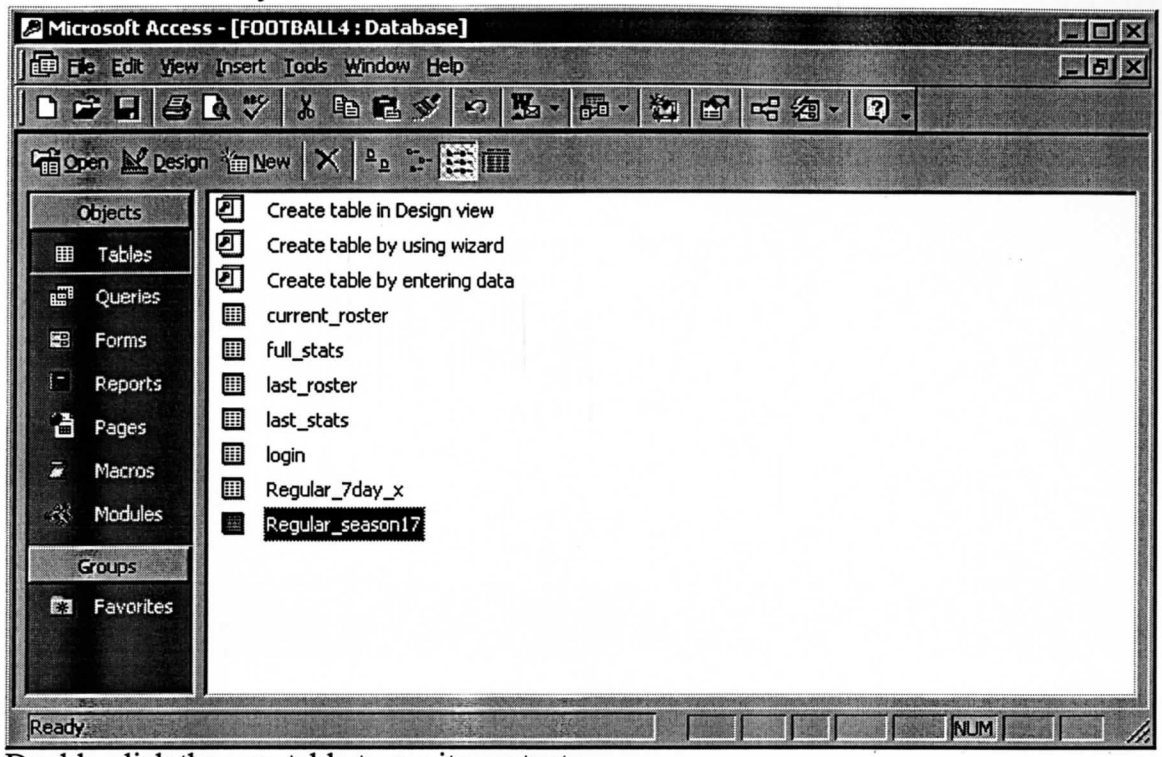


Here, chose the 2nd option button 'Choose my own primary key.' And be sure that the ID field is selected in the drop down menu. Press the 'Next' button to continue.

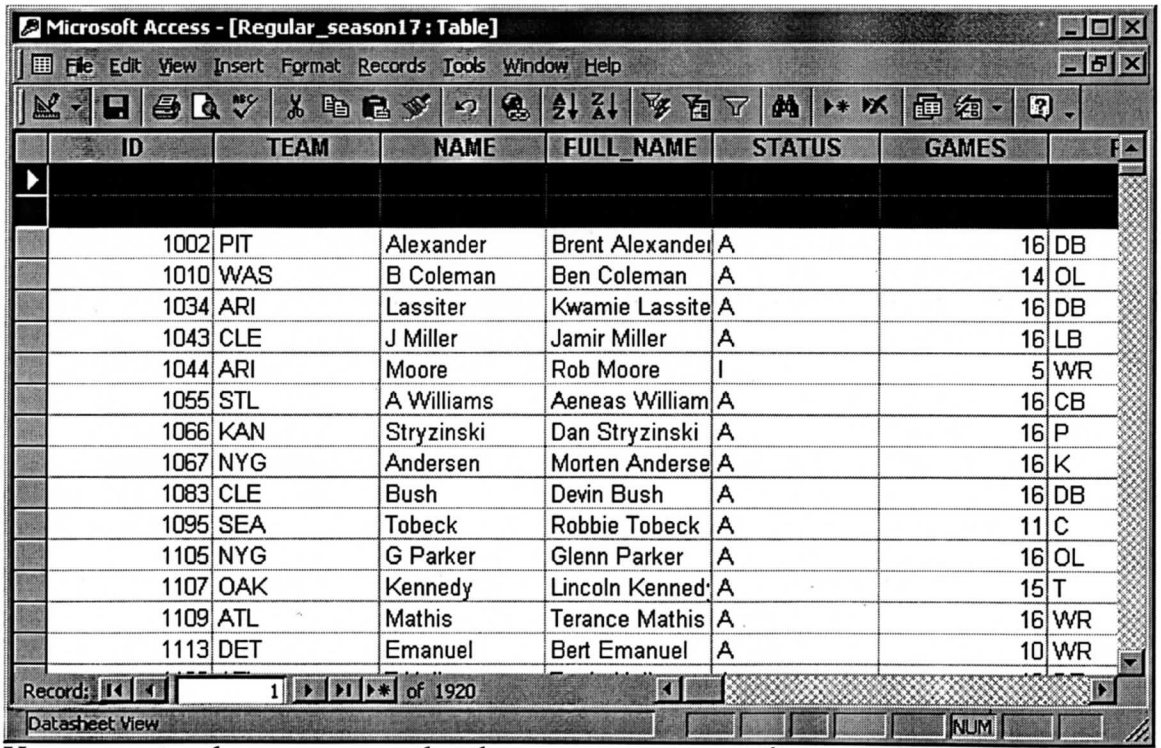


The last step of the wizard is to give the new table of data a name. Type a name in the text box and press the 'Finish' button to finish the wizard.

Now in MS Access you can see the table that we just imported along with the rest of the tables for the fantasy football database.



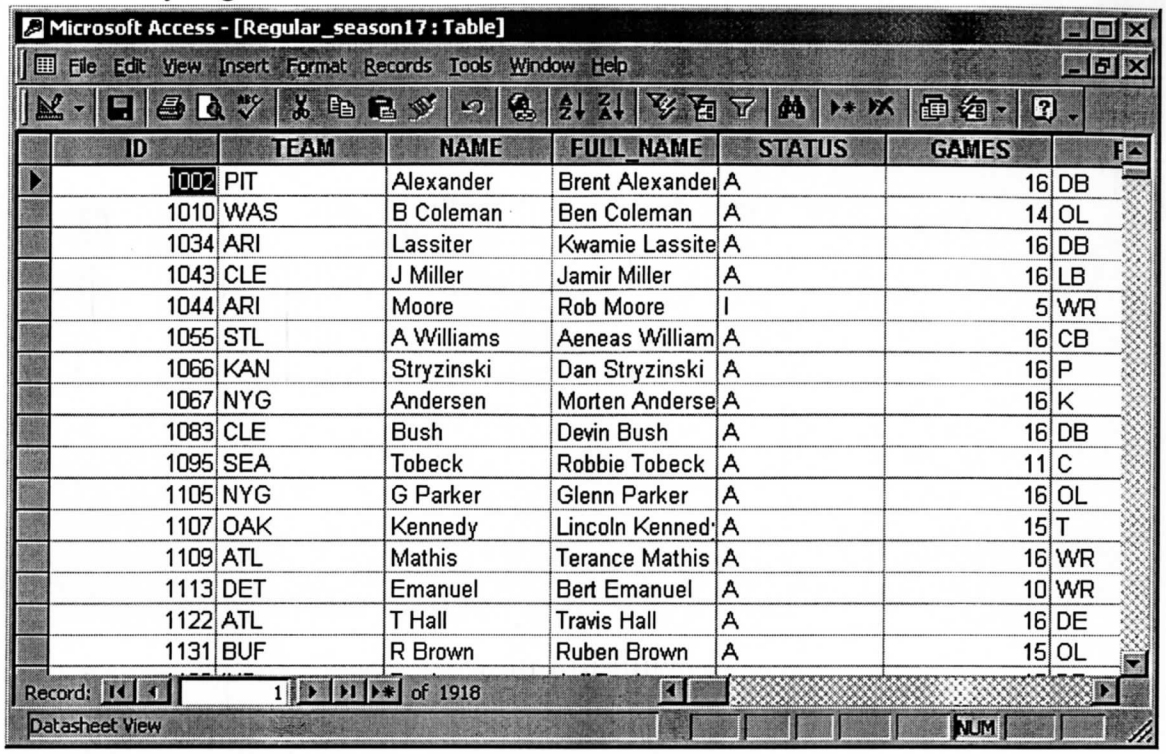
Double-click the new table to see its contents.



ID	TEAM	NAME	FULL_NAME	STATUS	GAMES	
1002	PIT	Alexander	Brent Alexander	A	16	DB
1010	WAS	B Coleman	Ben Coleman	A	14	OL
1034	ARI	Lassiter	Kwamie Lassite	A	16	DB
1043	CLE	J Miller	Jamir Miller	A	16	LB
1044	ARI	Moore	Rob Moore	I	5	WR
1055	STL	A Williams	Aeneas William	A	16	CB
1066	KAN	Stryzinski	Dan Stryzinski	A	16	P
1067	NYG	Andersen	Morten Anderse	A	16	K
1083	CLE	Bush	Devin Bush	A	16	DB
1095	SEA	Tobeck	Robbie Tobeck	A	11	C
1105	NYG	G Parker	Glenn Parker	A	16	OL
1107	OAK	Kennedy	Lincoln Kenned	A	15	T
1109	ATL	Mathis	Terance Mathis	A	16	WR
1113	DET	Emanuel	Bert Emanuel	A	10	WR

You may get and error message that there was an error creating a row in the table because the primary key cannot be NULL. This is because the text file had some blank lines at

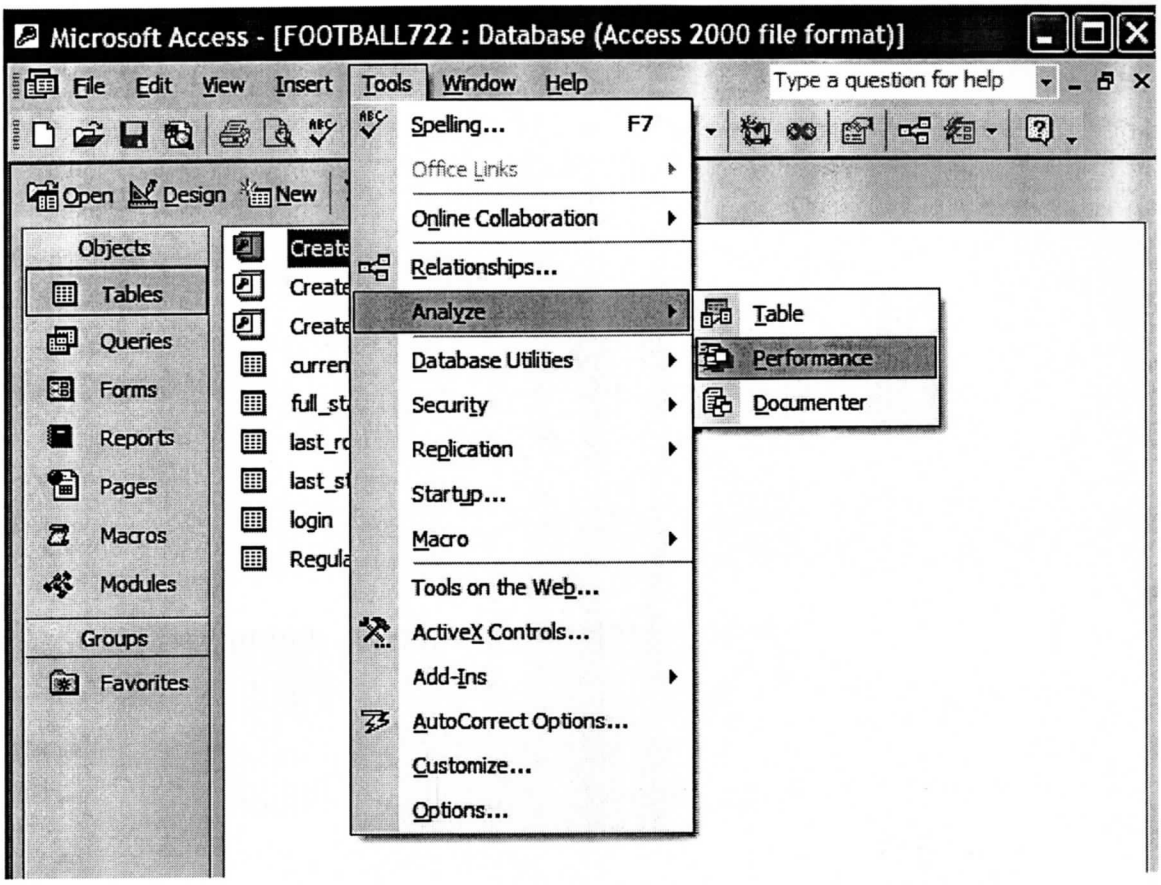
the bottom. To fix this problem simply delete the first two records in the table. Our new table is ready to go.



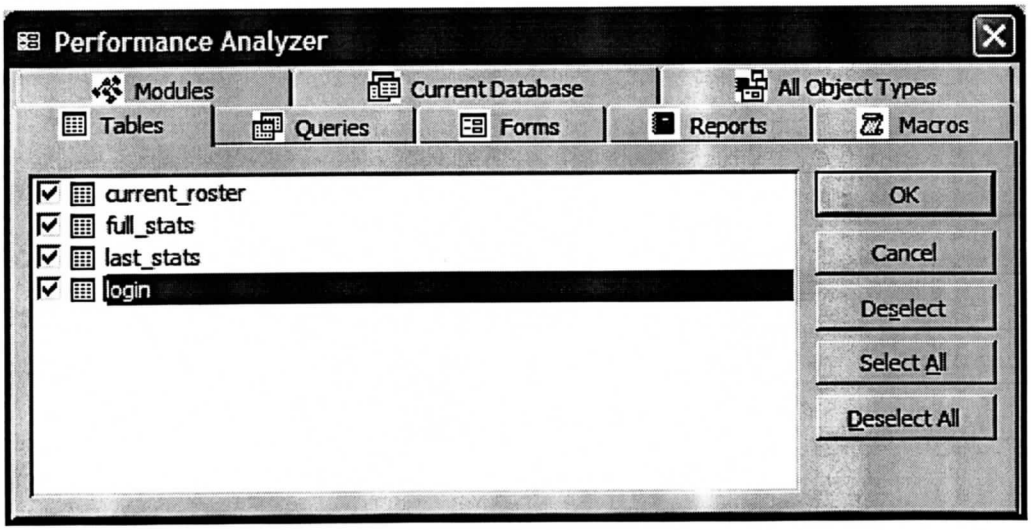
ID	TEAM	NAME	FULL_NAME	STATUS	GAMES	F
1002	PIT	Alexander	Brent Alexande	A	16	DB
1010	WAS	B Coleman	Ben Coleman	A	14	OL
1034	ARI	Lassiter	Kwamie Lassite	A	16	DB
1043	CLE	J Miller	Jamir Miller	A	16	LB
1044	ARI	Moore	Rob Moore	I	5	WR
1055	STL	A Williams	Aeneas William	A	16	CB
1066	KAN	Stryzinski	Dan Stryzinski	A	16	P
1067	NYG	Andersen	Morten Anderse	A	16	K
1083	CLE	Bush	Devin Bush	A	16	DB
1095	SEA	Tobeck	Robbie Tobeck	A	11	C
1105	NYG	G Parker	Glenn Parker	A	16	OL
1107	OAK	Kennedy	Lincoln Kenned	A	15	T
1109	ATL	Mathis	Terance Mathis	A	16	WR
1113	DET	Emanuel	Bert Emanuel	A	10	WR
1122	ATL	T Hall	Travis Hall	A	16	DE
1131	BUF	R Brown	Ruben Brown	A	15	OL

Database Optimization

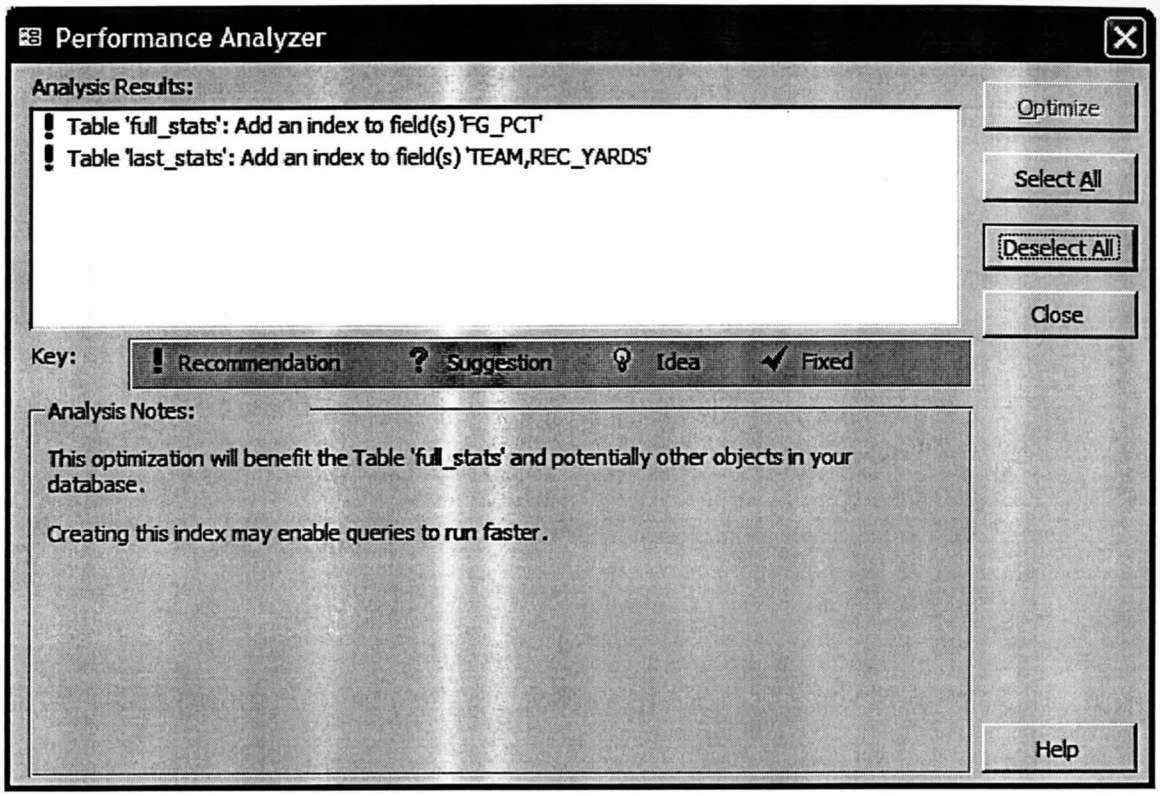
Select the 'Tools' menu, and then select 'Analyze' then 'Performance'



You will then see the 'Performance Analyzer' dialog box. Select the 'Tables' tab then select all of the tables in the database and press 'OK'

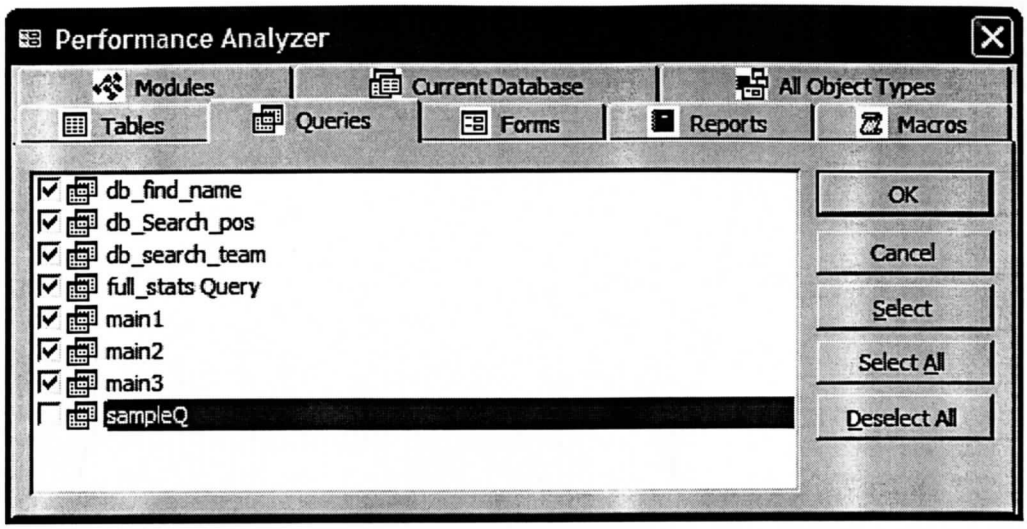


You will then see:

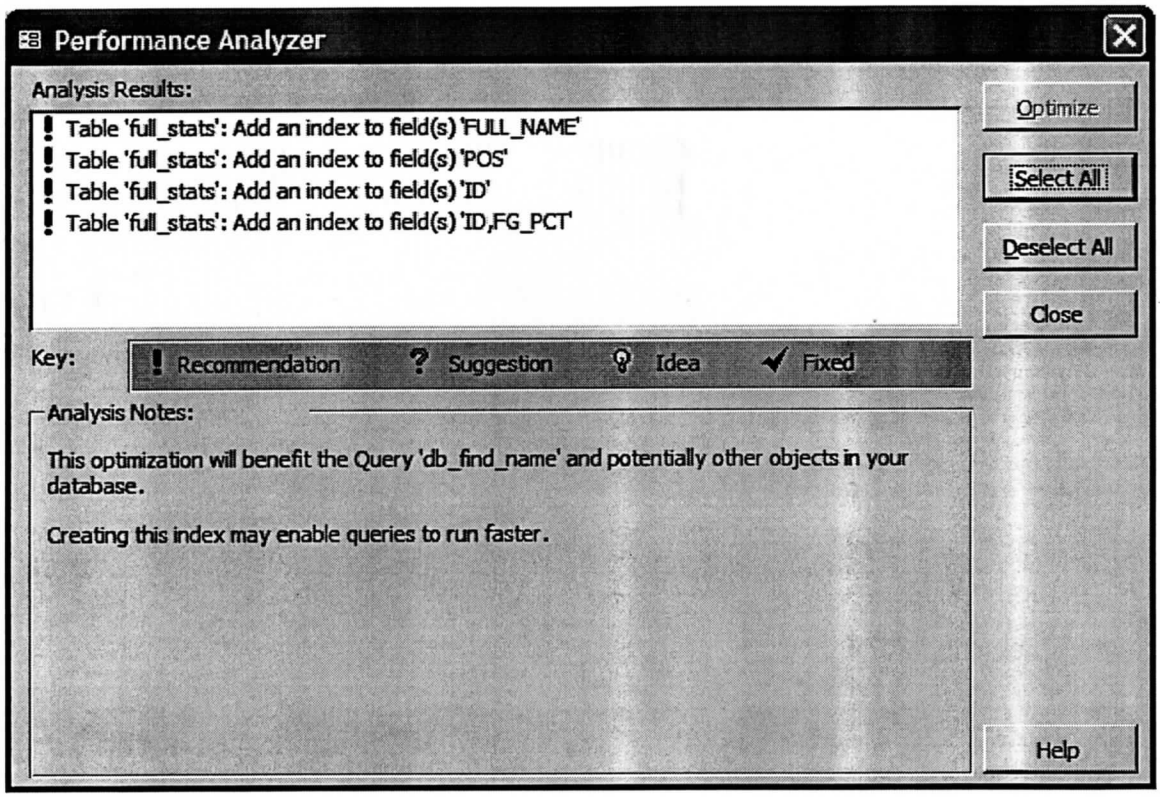


Access recommends adding indexes on several fields. Press the 'Select All' button and then the 'Optimize' button. Pressing the close button will then exit the Performance Analyzer.

I have also created the same queries that I use in the ASP script in Access in order to optimize the tables to handle the queries. So once again bring up the Performance Analyzer but this time select the 'Queries' tab and then select the checkbox by each of the created queries. When finished press OK.

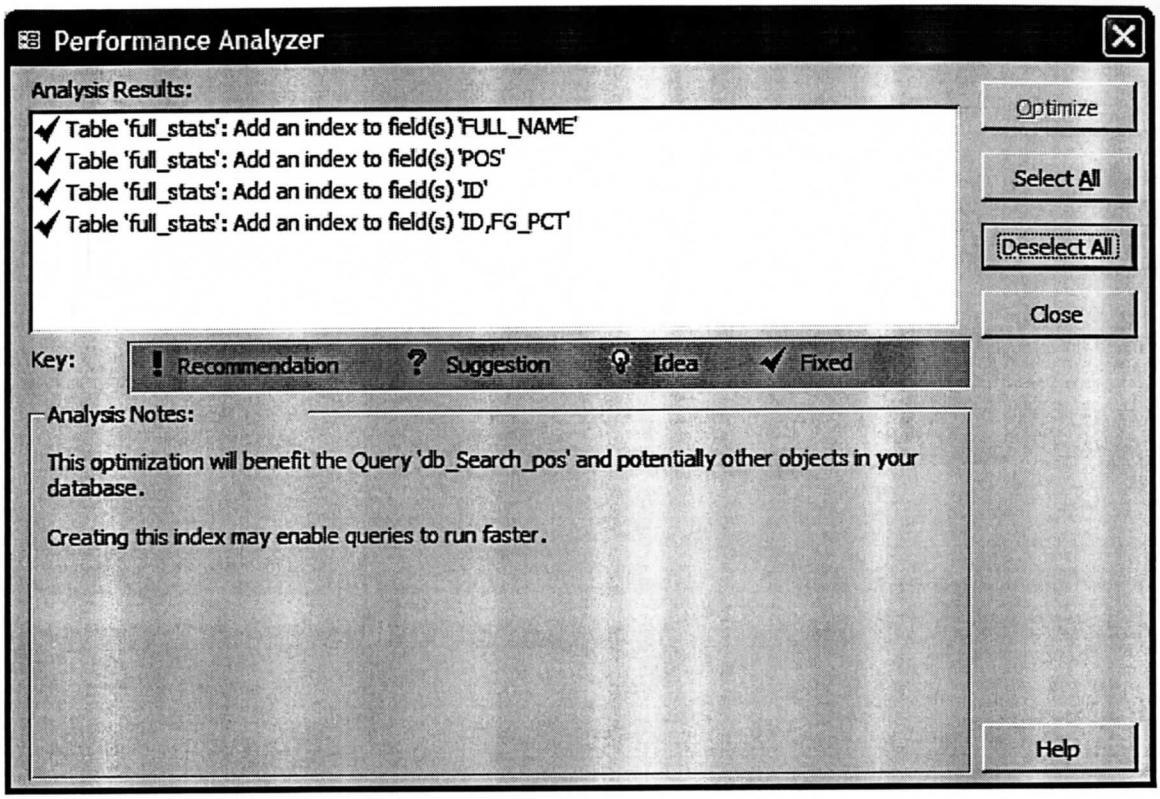


You will then see this screen:



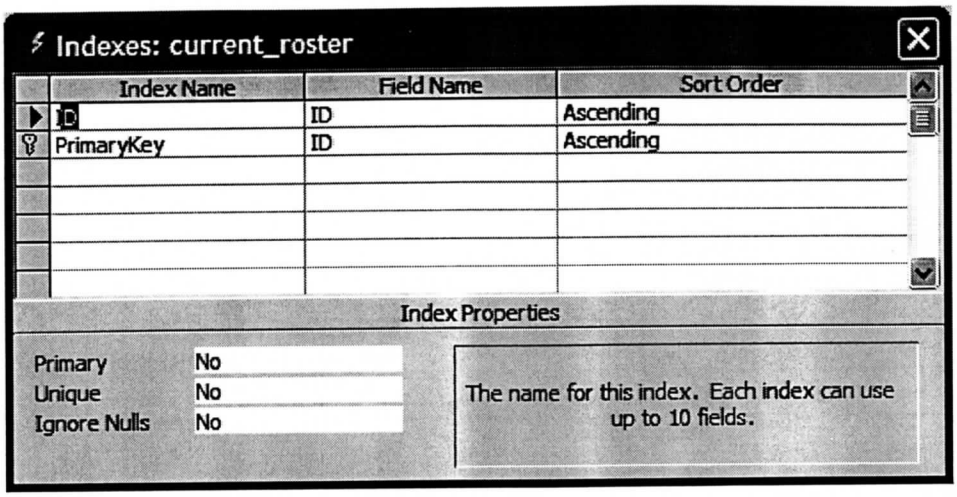
The Performance Analyzer has found four potential areas that could be streamlined to improve performance. You can click on each of the items to get a brief explanation of the analysis. All of these points just mention that an index will be built that should help

the speed of the queries. Once finished reading, press the 'Select All' button and then the 'Optimize' button and you will see:

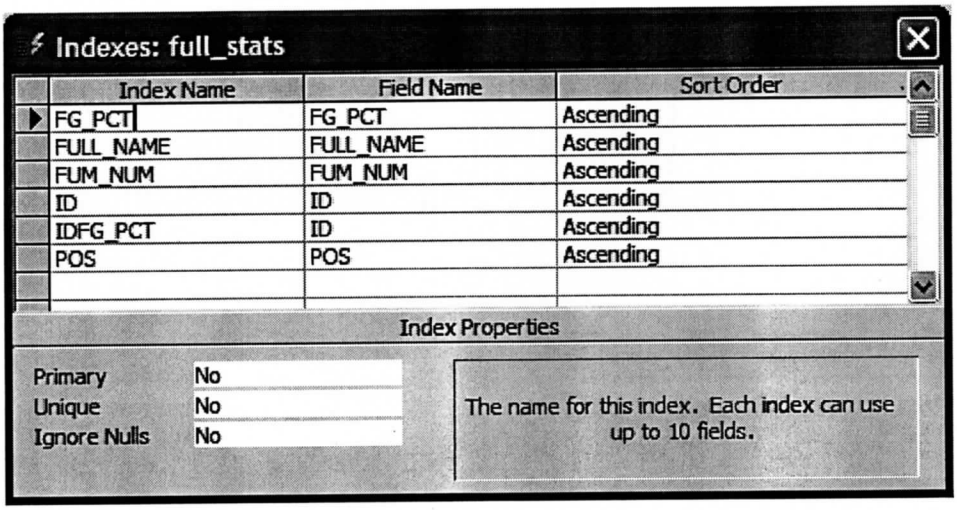


Now press the 'Close' button to finish.

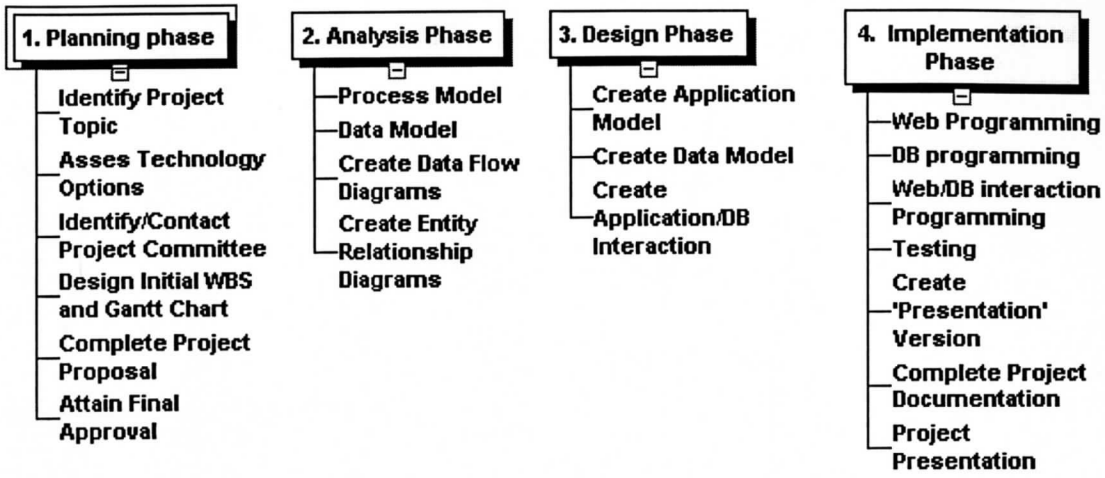
To view the indexes that Access created for you select the desired table, go to 'Design View' then select the 'View' menu and then select 'Indexes'. You should see something like this:



Above is the index listing for the current_roster table. As you can see the Performance Analyzer has added 2 indexes to this table. Below you can see the indexes on the full_stats table:

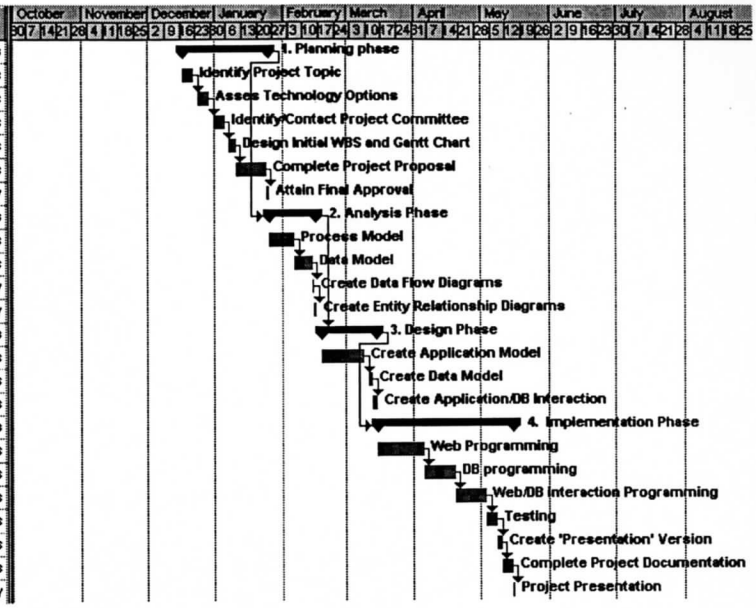


Appendix C: Work Breakdown Structure



Appendix D: Gantt Chart

	Task Name	Duration
1	**1. Planning phase**	**29 days**
2	Identify Project Topic	5 days
3	Asses Technology Options	5 days
4	Identify/Contact Project Committee	5 days
5	Design Initial WBS and Gantt Chart	3 days
6	Complete Project Proposal	10 days
7	Attain Final Approval	1 day
8	**2. Analysis Phase**	**16 days**
9	Process Model	8 days
10	Data Model	6 days
11	Create Data Flow Diagrams	1 day
12	Create Entity Relationship Diagrams	1 day
13	**3. Design Phase**	**19 days**
14	Create Application Model	15 days
15	Create Data Model	2 days
16	Create Application/DB Interaction	2 days
17	**4. Implementation Phase**	**44 days**
18	Web Programming	15 days
19	DB programming	10 days
20	Web/DB interaction Programming	10 days
21	Testing	3 days
22	Create 'Presentation' Version	2 days
23	Complete Project Documentation	3 days
24	Project Presentation	1 day



Appendix E: Sample Project Source Code

- Add_player3.asp – details of how the Access database is updated with SQL and ASP – p. E-1
- Admin_edit_users.asp – an additional example of updating the DN – p, E-3
- Main.asp – code for the team homepage – p.E-4
- Admin_update_stats.asp – most technical page with several select statements and update statements inside a loop. P. E-5

```
<!--
add_player3.asp
Jeremy Pauli - MSIS Project
-->
<%@ Language="VBScript" %>
<!-- #include file="ole.asp" -->
<!-- #include file="errors.asp" -->
<%
On Error Resume Next
'Dim Pos, numID
POS = request.QueryString("pos")
numID = request.QueryString("id")
Dim Conn       ' Connection object
Dim objProp      ' Property object
Dim objError     ' Error object
'********** DB CONNECTION STUFF ****************************************
        SQL = "Update current_roster SET "&POS&" = "&numID&", last_change ='"&NOW&"' where
ID = "&Session("ID")&";"

    Set Conn = Server.CreateObject("ADODB.Connection")

    Conn.Mode = 3     '3 = adModeReadWrite

Conn.Open "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=c:\inetpub\wwwroot\\fpdb\football.mdb;"

    Conn.Execute(SQL)

    CheckForErrors(Conn)

    Conn.Close
                response.redirect "main.asp"
%>
```

```
<!--
admin_edit_users.asp
Jeremy Pauli - MSIS Project
-->
<%@ Language="VBScript" %>
<!-- #include file="ole.asp" -->
<!-- #include file="functions.asp" -->
<!-- #include file="functions2.asp" -->
<%
'----SQL Query.....
                    strSQL = "Select ID, username, password, division from login where username <>
'admin' order by username;"
                    Set rstSearch = Server.CreateObject("ADODB.Recordset")
                    rstSearch.Open strSQL, cnnSearch, adOpenStatic, adLockReadOnly, adCmdText
%>
<hr class="bottom"><hr><br>
<center><font size=4>Select Users to Edit</font>
        <table align="center"><form method="post" action="admin_proc_edit.asp">
                    <th>X</th>
                    <th>ID</th>
                    <th>Username</th>
                    <th>Password</th>
                    <th>Division</th>
<%
count = 1
While Not rstSearch.eof
        %><tr class="<%=count mod 2%>">
                    <td><input type="Checkbox" value="ON" name="userID<%=
rstSearch("ID")%>"></td>
                    <td><%= rstSearch.Fields("ID") %></td>
                    <td><%= rstSearch.Fields("username") %></td>
                    <td><%= rstSearch.Fields("password") %></td>
                    <td><%= rstSearch.Fields("division") %></td>
                    </tr><%
                    count = count + 1
        rstSearch.movenext
WEND
%></table><input type ="submit" value="Edit Selected Users" name="submit"></form><!-- #include
file="admin_nav.asp" -->
```

```
<!--
main.asp
Jeremy Pauli - MSIS Project
-->
<meta name="Microsoft Border" content="none">
<%@ Language="VBScript" %>
<!-- #include file="ole.asp" -->
<!-- #include file="functions.asp" -->
<!-- #include file="functions2.asp" -->
<%
if (Session("username")="") THEN
        response.redirect"home.htm"
End IF
IF (Session("username")="admin") THEN
        response.redirect "admin_home.asp"
End IF
Dim bolPlayer
Dim freeze
freeze = false
strDate = Now()
'Lock the roster if statement - getting day of week - only edit roster on tuesday-
'Roster Edits can't be made on Sunday or Monday.
        'if ((weekday(strDate) = 1) OR (weekday(strDate) = 7)) Then
        '       freeze = true
        'END IF

'----1st SQL Query.....
                strSQL = "Select * from current_roster where ID = "&Session("ID")& ";"
                Set rstSearch = Server.CreateObject("ADODB.Recordset")
                rstSearch.Open strSQL, cnnSearch, adOpenStatic, adLockReadOnly, adCmdText
'----2nd SQL Query.....
                strSQL2 = "Select * from full_stats where ID = "&rstSearch.Fields("qb").Value&";"
                Set rstSearch2 = Server.CreateObject("ADODB.Recordset")
                rstSearch2.Open strSQL2, cnnSearch, adOpenStatic, adLockReadOnly, adCmdText
'----3rd SQL Query.....
                strSQL3 = "Select * from full_stats where ID = "&rstSearch.Fields("rb1").Value&";"
                Set rstSearch3 = Server.CreateObject("ADODB.Recordset")
                rstSearch3.Open strSQL3, cnnSearch, adOpenStatic, adLockReadOnly, adCmdText
'----4th SQL Query.....
                strSQL4 = "Select * from full_stats where ID = "&rstSearch.Fields("rb2").Value&";"
                Set rstSearch4 = Server.CreateObject("ADODB.Recordset")
                rstSearch4.Open strSQL4, cnnSearch, adOpenStatic, adLockReadOnly, adCmdText
'----5th SQL Query.....
                strSQL5 = "Select * from full_stats where ID = "&rstSearch.Fields("wr1").Value&";"
                Set rstSearch5 = Server.CreateObject("ADODB.Recordset")
                rstSearch5.Open strSQL5, cnnSearch, adOpenStatic, adLockReadOnly, adCmdText
'----6th SQL Query.....
                strSQL6 = "Select * from full_stats where ID = "&rstSearch.Fields("wr2").Value&";"
                Set rstSearch6 = Server.CreateObject("ADODB.Recordset")
                rstSearch6.Open strSQL6, cnnSearch, adOpenStatic, adLockReadOnly, adCmdText
'----7th SQL Query.....
                strSQL7 = "Select * from full_stats where ID = "&rstSearch.Fields("k").Value&";"
                Set rstSearch7 = Server.CreateObject("ADODB.Recordset")
                rstSearch7.Open strSQL7, cnnSearch, adOpenStatic, adLockReadOnly, adCmdText
'----8th SQL Query.....
```

```
                strSQL8 = "Select * from login where ID = "&Session("ID")& ";"
                Set rstSearch8 = Server.CreateObject("ADODB.Recordset")
                rstSearch8.Open strSQL8, cnnSearch, adOpenStatic, adLockReadOnly, adCmdText
%>
<!-- #include file="insert_info.asp" -->
<body style="text-align: left">

<tr>
<td colspan="2">
<font size="5">------------------------<font color="red"><%= Session("username")%></font> Team Home
</font>
</b>
</td>
</tr>
<!-- #include file="new_nav.asp" -->
<td>
<blockquote>
<br><b><font size="4">Your Team:</font> <br>

<%
'if freeze = true then
'       response.write"<font color=red>ROSTER IS FROZEN!</font>"
'End If
%>

<table border="1">
        <tr>
                <th>Pos.</th>
                <th>Name</th>
                <th>ID</th>
                <th> Total Points  </th>
                <th> Points/Game  </th>
                </tr>
        <tr class="1">
                <td align="center">QB</td>
                <td><%
                if (NOT(rstSearch2.eof)) THEN
                        response.write rstSearch2.Fields("full_name").Value
                        bolPlayer = true
                        else
                         response.write "<a href=db_search_pos.asp?search=QB><font size=3>Select a
QB</a>"
                         bolPlayer = False
                         End iF
                         %> </td>
                <td align="center"><%= create_link(rstSearch.Fields("qb").Value)  %> </td>
                <td align="center">
                <%
                IF (bolPlayer = true) THEN
                                'Getting values to compute points
                rush_yards = rstSearch2.Fields("rush_yards").Value
                pass_yards = rstSearch2.Fields("pass_yards").Value
                rush_td =  rstSearch2.Fields("rush_td").Value
                pass_td = rstSearch2.Fields("pass_td").Value
                ints = rstSearch2.Fields("int_thrown").Value
                fum = rstSearch2.Fields("fum_lost").Value
```

```
                twoptR = rstSearch2.Fields("2ptcon_run").Value
                twoptP = rstSearch2.Fields("2ptcon_pass").Value
                games = rstSearch2.Fields("games").Value
                %><%=compute_total_qb(pass_yards, rush_yards, rush_td, pass_td, ints, fum, twpptR,
twpptP, games)
                %><%
                Else %>
                 -</td><td align="center">-</td><%End IF %></td></tr>
                <tr class="2">
                <td align="center">RB1</td>
                <td><%
                if (NOT(rstSearch3.eof)) THEN
                 response.write rstSearch3.Fields("full_name").Value
                 bolPlayer=true
                 Else
                 response.write"<a href=db_search_pos.asp?SEARCH=RB><font size=3>Select a
RB1</a>"
                 bolPlayer=flase
                 End IF
                 %> </td>
                <td align="center"><% create_link(rstSearch.Fields("rb1").Value) %> </td>
                <td align="center">
                <%
                IF (bolPlayer = true) THEN
                'Getting values to compute points for---------- RB1
                rush_yards = rstSearch3.Fields("rush_yards").Value
                rec_yards = rstSearch3.Fields("rec_yards").Value
                rush_td =  rstSearch3.Fields("rush_td").Value
                rec_td = rstSearch3.Fields("rec_td").Value
                fum = rstSearch3.Fields("fum_lost").Value
                twoptR = rstSearch3.Fields("2ptcon_run").Value
                twoptRec = rstSearch3.Fields("2ptcon_rec").Value
                games = rstSearch3.Fields("games").Value
                %>
                <%=compute_total_rb(rec_yards, rush_yards, rush_td, rec_td, fum, twoptR, twoptrec,
games) %><%
                Else %>
                 -</td><td align="center">-</td><%End IF %></td>
        </tr>
                <tr class="1">
                <td align="center">RB2</td>
                <td><%
                if (NOT(rstSearch4.eof)) THEN
                 response.write rstSearch4.Fields("full_name").Value
                 bolPlayer=true
                 Else
                 response.write"<a href=db_search_pos.asp?search=RB><font size=3>Select a RB2</a>"
                 bolPlayer=flase
                 End IF
                 %> </td>
                <td align="center"><% create_link(rstSearch.Fields("rb2").Value) %> </td>
                <td align="center">
                <%
                IF (bolPlayer = true) THEN
                'Getting values to compute points for---------- RB1
                rush_yards = rstSearch4.Fields("rush_yards").Value
```

```
                rec_yards = rstSearch4.Fields("rec_yards").Value
                rush_td =  rstSearch4.Fields("rush_td").Value
                rec_td = rstSearch4.Fields("rec_td").Value
                fum = rstSearch4.Fields("fum_lost").Value
                twoptR = rstSearch4.Fields("2ptcon_run").Value
                twoptRec = rstSearch4.Fields("2ptcon_rec").Value
                games = rstSearch4.Fields("games").Value
                %>
                <%=compute_total_rb(rec_yards, rush_yards, rush_td, rec_td, fum, twoptR, twoptrec,
games) %><% Else %>
                 -</td><td align="center">-</td><%End IF %></td>
<!--
*****************************************************************************************
*********-->

        </tr><tr class="2">
                <td align="center">WR1</td>
                <td><% if (NOT(rstSearch5.eof)) then
                response.write rstSearch5.Fields("full_name").Value
                bolPlayer=true
                 Else
                 response.write"<a href=db_search_pos.asp?search=WR><font size=3>Select a
WR1</a>"
                 bolPlayer=flase
                 End IF  %> </td>
                <td align="center"><% create_link(rstSearch.Fields("wr1").Value) %> </td>
                <td align="center"><%
                IF (bolPlayer = true) THEN
                'Getting values to compute points for---------- wr1
                rush_yards = rstSearch5.Fields("rush_yards").Value
                rec_yards = rstSearch5.Fields("rec_yards").Value
                rush_td =  rstSearch5.Fields("rush_td").Value
                rec_td = rstSearch5.Fields("rec_td").Value
                fum = rstSearch5.Fields("fum_lost").Value
                twoptR = rstSearch5.Fields("2ptcon_run").Value
                twoptRec = rstSearch5.Fields("2ptcon_rec").Value
                games = rstSearch5.Fields("games").Value
                %>
                <%=compute_total_rb(rec_yards, rush_yards, rush_td, rec_td, fum, twoptR, twoptrec,
games)%><%
                Else %>
                 -</td><td align="center">-</td><%End IF %></td>
        </tr>
        <tr class="1">
                <td align="center">WR2</td>
                <td><%
                IF (NOt(rstSearch6.eof)) THEN
                response.write rstSearch6.Fields("full_name").Value
                bolPlayer=true
                 Else
                 response.write"<a href=db_search_pos.asp?search=WR><font size=3>Select a
WR2</a>"
                 bolPlayer=flase
                 End IF  %> </td>
                <td align="center"><% create_link(rstSearch.Fields("wr2").Value) %> </td>
                <td align="center"><%
```

```
                    if (bolPlayer = true) THEN
                    'Getting values to compute points for---------- wr2
                    rush_yards = rstSearch6.Fields("rush_yards").Value
                    rec_yards = rstSearch6.Fields("rec_yards").Value
                    rush_td =  rstSearch6.Fields("rush_td").Value
                    rec_td = rstSearch6.Fields("rec_td").Value
                    fum = rstSearch6.Fields("fum_lost").Value
                    twoptR = rstSearch6.Fields("2ptcon_run").Value
                    twoptRec = rstSearch6.Fields("2ptcon_rec").Value
                    games = rstSearch6.Fields("games").Value
                    %>
                    <%=compute_total_rb(rec_yards, rush_yards, rush_td, rec_td, fum, twoptR, twoptrec,
games)%><%
                    Else %>
                     -</td><td align="center">-</td><%End IF %>
          </tr>
          <tr class="2">
                    <td align="center">K</td>
                    <td><%
                    if (NOT(rstSearch7.eof))THEN
                    response.write rstSearch7.Fields("full_name").Value
                    bolPlayer=true
                     Else
                     response.write"<a href=db_search_pos.asp?search=K><font size=3>Select a K</a>"
                     bolPlayer=flase
                     End IF
                    %> </td>
                    <td align="center"><% create_link(rstSearch.Fields("k").Value) %> </td>
                    <td align="center">
                    <%
                    If (bolPlayer = true) THEN
                    'Getting values to compute points for---------- K
                    fg_made = rstSearch7.Fields("fg_made").Value
                    fg_missed = rstSearch7.Fields("fg_att").Value - rstSearch7.Fields("fg_made").Value
                    pat_made = rstSearch7.Fields("epts_made").Value
                    pat_missed = rstSearch7.Fields("epts_att").Value - rstSearch7.Fields("epts_made").Value
                    games = rstSearch7.Fields("games").Value
                    %>
                    <%=compute_total_k(fg_made, pat_made, fg_missed, pat_missed, games) %><%
                    Else %>
                     -</td><td align="center">-</td><%End IF %>
</td>
          </tr>
          </table>
          <font size="2">        </font>
          <font size="1" color="#808080">Last Roster Change: <%= rstSearch.Fields("last_change").Value
%> </font>
<br>

<b><font size="4"><br>
Scores by Week </font> </b>
  <div align="left">
    <table border="1">
      <tr>
        <th align="center"><b>WEEK</font></b></td>
  </b>
```

```
    </td>
    <th><b>SCORE</b></td>
    <th><b>WEEK</font></b></td>
</b>
    </td>
    <th><b>SCORE</b></td>
   </tr>
   <tr>
    <th align="center"><b>1</b></td>
    <td align="center" bgcolor="#FFFFFF"><%=rstSearch8.Fields("wk1").value%> </td>
    <th align="center"><b>10</b></td>
    <td align="center" bgcolor="#FFFFFF"><%=rstSearch8.Fields("wk10").value%> </td>
   </tr>
   <tr>
    <th align="center"><b>2</b></td>
    <td align="center" bgcolor="#FFFFFF"><%=rstSearch8.Fields("wk2").value%> </td>
    <th align="center"><b>11</b></td>
    <td align="center" bgcolor="#FFFFFF"><%=rstSearch8.Fields("wk11").value%> </td>
   </tr>
   <tr>
    <th align="center"><b>3</b></td>
    <td align="center" bgcolor="#FFFFFF"><%=rstSearch8.Fields("wk3").value%> </td>
    <th align="center"><b>12</b></td>
    <td align="center" bgcolor="#FFFFFF"><%=rstSearch8.Fields("wk12").value%> </td>
   </tr>
   <tr>
    <th align="center"><b>4</b></td>
    <td align="center" bgcolor="#FFFFFF"><%=rstSearch8.Fields("wk4").value%> </td>
    <th align="center"><b>13</b></td>
    <td align="center" bgcolor="#FFFFFF"><%=rstSearch8.Fields("wk13").value%> </td>
   </tr>
   <tr>
    <th align="center"><b>5</b></td>
    <td align="center" bgcolor="#FFFFFF"><%=rstSearch8.Fields("wk5").value%> </td>
    <th align="center"><b>14</b></td>
    <td align="center" bgcolor="#FFFFFF"><%=rstSearch8.Fields("wk14").value%> </td>
   </tr>
   <tr>
    <th align="center"><b>6</b></td>
    <td align="center" bgcolor="#FFFFFF"><%=rstSearch8.Fields("wk6").value%> </td>
    <th align="center"><b>15</b></td>
    <td align="center" bgcolor="#FFFFFF"><%=rstSearch8.Fields("wk15").value%> </td>
   </tr>
   <tr>
    <th align="center"><b>7</b></td>
    <td align="center" bgcolor="#FFFFFF"><%=rstSearch8.Fields("wk7").value%> </td>
    <th align="center"><b>16</b></td>
    <td align="center" bgcolor="#FFFFFF"><%=rstSearch8.Fields("wk16").value%> </td>
   </tr>
   <tr>
    <th align="center"><b>8</b></td>
    <td align="center" bgcolor="#FFFFFF"><%=rstSearch8.Fields("wk8").value%> </td>
    <th align="center"><b>17</b></td>
    <td align="center" bgcolor="#FFFFFF"><%=rstSearch8.Fields("wk17").value%> </td>
   </tr>
   <tr>
```

```
      <th align="center"><b>9</b></td>
      <td align="center" bgcolor="#FFFFFF"><%=rstSearch8.Fields("wk9").value%> </td>
      <th align="center"><font size="4">TOTAL</font></td>
      <td align="center" bgcolor="#FFFFFF"><b><%=rstSearch8.Fields("total").value%> </b></td>
    </tr>
  </table>
</div>
 <br>
 <p>
 </font>
 </b>
 </blockquote>
 <!-- #include file="bottom_nav.asp" -->
```

```
<%@ Language="VBScript" %>
<!-- #include file="ole.asp" -->
<!-- #include file="functions2.asp" -->
<!-- #include file="errors.asp" -->
<%
on error resume next
'*****************************************************************************************
**********
'getting week number from form.
dim weekNum
weekNum = "wk"& request.querystring("weekNum")
' page that will update the weekly statistics, update each users score for the week, ect....

'LOOP THROUGH EACH USER AND TOTAL EACH PLAYER'S WEEKLY TOTAL AND SUM IT
AND ADD IT TO THE
'LOGIN TABLE
'Statement that takes all rosters and user ID's and puts them into a record set
'to be looped through --> 1 loop for each user.
'Each loop contains 10 SQL queries. 2 updates and 8 selects
'SQL statement
                strSQL = "Select ID, QB, RB1, RB2, WR1, WR2, K from current_roster where ID <>5;"
'Execute
                Set rstSearch = Server.CreateObject("ADODB.Recordset")

'Open our recordset
                rstSearch.Open strSQL, cnnSearch, 3, 1, &H0001
'************************************
'1.  getting each users current roster
'-------------------------------------
Do While Not rstSearch.EOF
        %>
        <%
        dim week1, week2, week3, week4, week5, week6, week7, week8, week9, week10, week11,
week12, week13, week14, week15, week16, week17
        dim week1a, week2a, week3a, week4a, week5a, week6a, week7a, week8a, week9a, week10a,
week11a, week12a, week13a, week14a, week15a, week16a, week17a
        datotal = 0
        rb1_total = 0
        rb2_total = 0
        wr1_total = 0
        wr2_total = 0
        k_total = 0
        week_total = 0
        user_grand_total = 0
        if (rstSearch.Fields("qb").Value < 1) THEN
                'do not do anything....
        Else
        '************************************
        'Then add up each players points
        '1 sql statement for each player
        'SQL for QB
        '-------------------------------------
        strSQL2 = "Select * from last_stats where ID = "&rstSearch.Fields("qb").Value&";"
        Set rstSearch2 = Server.CreateObject("ADODB.Recordset")
        rstSearch2.Open strSQL2, cnnSearch, adOpenStatic, adLockReadOnly, adCmdText
        rush_yards = rstSearch2.Fields("rush_yards").Value
```

```
pass_yards = rstSearch2.Fields("pass_yards").Value
rush_td =  rstSearch2.Fields("rush_td").Value
pass_td = rstSearch2.Fields("pass_td").Value
ints = rstSearch2.Fields("int_thrown").Value
fum = rstSearch2.Fields("fum_lost").Value
twoptR = rstSearch2.Fields("2ptcon_run").Value
twoptP = rstSearch2.Fields("2ptcon_pass").Value
games = rstSearch2.Fields("games").Value
datotal = (pass_yards*1)+(rush_yards*2)+(pass_td*50)+(rush_td*75)-(int_thrown*50)-
(fum_lost*50)+(ptcon_run*25)+(ptcon_pass*25)
Set rstSearch3 = nothing
'----------
END IF
'------------
'*************************************************
'for RB1
if (rstSearch.Fields("rb1").Value < 1) THEN
        'do not do anything....
Else
strSQL3 = "Select * from last_stats where ID = "&rstSearch.Fields("rb1").Value&";"
Set rstSearch3 = Server.CreateObject("ADODB.Recordset")
rstSearch3.Open strSQL3, cnnSearch, adOpenStatic, adLockReadOnly, adCmdText
rush_yards = rstSearch3.Fields("rush_yards").Value
        rec_yards = rstSearch3.Fields("rec_yards").Value
        rush_td =  rstSearch3.Fields("rush_td").Value
        rec_td = rstSearch3.Fields("rec_td").Value
        fum = rstSearch3.Fields("fum_lost").Value
        twoptR = rstSearch3.Fields("2ptcon_run").Value
        twoptRec = rstSearch3.Fields("2ptcon_rec").Value
        games = rstSearch3.Fields("games").Value
        rb1_total =
(rec_yards*rec_yard)+(rush_yards*rush_yard)+(rec_td*td_rec)+(rush_td*td_rush)-
(fum_lost*fumble)+(ptcon_run*rush_2)+(ptcon_rec*rec_2)
        Set rstSearch3 = nothing
        '----------
        END IF
        '------------
'*************************************************
'end RB1

'*************************************************
'For RB2
if (rstSearch.Fields("rb2").Value < 1) THEN
        'do not do anything....
Else
strSQL4 = "Select * from last_stats where ID = "&rstSearch.Fields("rb2").Value&";"
Set rstSearch4 = Server.CreateObject("ADODB.Recordset")
rstSearch4.Open strSQL4, cnnSearch, adOpenStatic, adLockReadOnly, adCmdText
rush_yards = rstSearch4.Fields("rush_yards").Value
        rec_yards = rstSearch4.Fields("rec_yards").Value
        rush_td =  rstSearch4.Fields("rush_td").Value
        rec_td = rstSearch4.Fields("rec_td").Value
        fum = rstSearch4.Fields("fum_lost").Value
        twoptR = rstSearch4.Fields("2ptcon_run").Value
        twoptRec = rstSearch4.Fields("2ptcon_rec").Value
        games = rstSearch4.Fields("games").Value
```

```
                rb2_total =
(rec_yards*rec_yard)+(rush_yards*rush_yard)+(rec_td*td_rec)+(rush_td*td_rush)-
(fum_lost*fumble)+(ptcon_run*rush_2)+(ptcon_rec*rec_2)
                Set rstSearch4 = nothing
                '-----------
                END IF
                '------------
                '*************************************************
        'For WR1
        if (rstSearch.Fields("wr1").Value < 1) THEN
                'do not do anything....
        Else
        strSQL5 = "Select * from last_stats where ID = "&rstSearch.Fields("wr1").Value&";"
        Set rstSearch5 = Server.CreateObject("ADODB.Recordset")
        rstSearch5.Open strSQL5, cnnSearch, adOpenStatic, adLockReadOnly, adCmdText
        rush_yards = rstSearch5.Fields("rush_yards").Value
                rec_yards = rstSearch5.Fields("rec_yards").Value
                rush_td =  rstSearch5.Fields("rush_td").Value
                rec_td = rstSearch5.Fields("rec_td").Value
                fum = rstSearch5.Fields("fum_lost").Value
                twoptR = rstSearch5.Fields("2ptcon_run").Value
                twoptRec = rstSearch5.Fields("2ptcon_rec").Value
                games = rstSearch5.Fields("games").Value
                wr1_total =
(rec_yards*rec_yard)+(rush_yards*rush_yard)+(rec_td*td_rec)+(rush_td*td_rush)-
(fum_lost*fumble)+(ptcon_run*rush_2)+(ptcon_rec*rec_2)
                Set rstSearch5 = nothing
                '-----------
                END IF
                '------------
        '*************************************************************************
        'For WR2
        '*************************************************************************
        if (rstSearch.Fields("wr2").Value < 1) THEN
                'do not do anything....
        Else
        strSQL6 = "Select * from last_stats where ID = "&rstSearch.Fields("wr2").Value&";"
        Set rstSearch6 = Server.CreateObject("ADODB.Recordset")
        rstSearch6.Open strSQL6, cnnSearch, adOpenStatic, adLockReadOnly, adCmdText
        rush_yards = rstSearch6.Fields("rush_yards").Value
                rec_yards = rstSearch6.Fields("rec_yards").Value
                rush_td =  rstSearch6.Fields("rush_td").Value
                rec_td = rstSearch6.Fields("rec_td").Value
                fum = rstSearch6.Fields("fum_lost").Value
                twoptR = rstSearch6.Fields("2ptcon_run").Value
                twoptRec = rstSearch6.Fields("2ptcon_rec").Value
                games = rstSearch6.Fields("games").Value
                wr2_total =
(rec_yards*rec_yard)+(rush_yards*rush_yard)+(rec_td*td_rec)+(rush_td*td_rush)-
(fum_lost*fumble)+(ptcon_run*rush_2)+(ptcon_rec*rec_2)
                Set rstSearch6 = nothing
                '-----------
                END IF
                '------------
        '*************************************************************************
        'For K
```

```
'*************************************************************************
if (rstSearch.Fields("k").Value < 1) THEN
        'do not do anything....
Else
strSQL7 = "Select * from last_stats where ID = "&rstSearch.Fields("k").Value&";"
Set rstSearch7 = Server.CreateObject("ADODB.Recordset")
rstSearch7.Open strSQL7, cnnSearch, adOpenStatic, adLockReadOnly, adCmdText
        fg_made = rstSearch7.Fields("fg_made").Value
        fg_missed = rstSearch7.Fields("fg_att").Value - rstSearch7.Fields("fg_made").Value
        pat_made = rstSearch7.Fields("epts_made").Value
        pat_missed = rstSearch7.Fields("epts_att").Value - rstSearch7.Fields("epts_made").Value
        games = rstSearch7.Fields("games").Value
k_total = ((fg_made*FG)+(pat_made*pat))-((mpat*miss_pat)-(mfg*miss_fg))

Set rstSearch7 = nothing
'-----------
END IF
'-----------

'*************************************************************************
        'then update the login db with the new value for week total
        begin tran
        '-------------------------------------------------------------------
        week_total = datotal + rb1_total + rb2_total + wr1_total + wr2_total + k_total
        '-------------------------------------------------------------------
        SQL = "Update login SET "&weekNum&" = "&week_total&" where ID =
"&rstSearch.Fields("id").Value&";"
        '-------------------------------------------------------------------
Set Conn = Server.CreateObject("ADODB.Connection")
Conn.Mode = 3     '3 = adModeReadWrite
        Conn.Open "Provider=Microsoft.Jet.OLEDB.4.0;Data
Source=c:\inetpub\wwwroot\fpdb\football.mdb;"
Conn.Execute(SQL)
CheckForErrors(Conn)
Conn.Close
set Conn = nothing
        'For each user add up the total of each week set the value of the total field.
strSQL8 = "Select * from login where ID = "&rstSearch.Fields("id").Value&";"
Set rstSearch8 = Server.CreateObject("ADODB.Recordset")
rstSearch8.Open strSQL8, cnnSearch, adOpenDynamic, adLockReadOnly, adCmdText
name = rstSearch8.Fields("username").Value
week1 = rstSearch8.Fields("wk1").Value
week2 = rstSearch8.Fields("wk2").Value
week3 = rstSearch8.Fields("wk3").Value
week4 = rstSearch8.Fields("wk4").Value
week5 = rstSearch8.Fields("wk5").Value
week6 = rstSearch8.Fields("wk6").Value
week7 = rstSearch8.Fields("wk7").Value
week8 = rstSearch8.Fields("wk8").Value
week9 = rstSearch8.Fields("wk9").Value
week10 = rstSearch8.Fields("wk10").Value
week11 = rstSearch8.Fields("wk11").Value
week12 = rstSearch8.Fields("wk12").Value
week13 = rstSearch8.Fields("wk13").Value
week14 = rstSearch8.Fields("wk14").Value
week15 = rstSearch8.Fields("wk15").Value
```

```
        week16 = rstSearch8.Fields("wk16").Value
        week17 = rstSearch8.Fields("wk17").Value
        user_grand_total = week1 +
week2+week3+week4+week5+week6+week7+week8+week9+week10+week11+week12+week13+week14
+week15+week16+week17
'-------------------------------------------------------------------------------------------
'Now update the DB with the new Grand Total - same as updateing the weekly total...
        SQL = "Update login SET total = "&user_grand_total&" where ID =
"&rstSearch.Fields("id").Value&";"
        Set Conn2 = Server.CreateObject("ADODB.Connection")
        Conn2.Mode = 3     '3 = adModeReadWrite
                Conn2.Open "Provider=Microsoft.Jet.OLEDB.4.0;Data
Source=c:\inetpub\wwwroot\fpdb\football.mdb;"
        Conn2.Execute(SQL)
        CheckForErrors(Conn2)
        Conn2.Close
        set Conn2 = nothing
        commit tran
'*******************************************************************************************
*****************
'       now do a new query to get the new values out of the db!!!
        strSQL9 = "Select * from login where ID = "&rstSearch.Fields("id").Value&";"
        Set rstSearch9 = Server.CreateObject("ADODB.Recordset")
        rstSearch9.Open strSQL9, cnnSearch, adOpenDynamic, adLockReadonly, adCmdText
        week1a = rstSearch9.Fields("wk1").Value
        week2a = rstSearch9.Fields("wk2").Value
        week3a = rstSearch9.Fields("wk3").Value
        week4a = rstSearch9.Fields("wk4").Value
        week5a = rstSearch9.Fields("wk5").Value
        week6a = rstSearch9.Fields("wk6").Value
        week7a = rstSearch9.Fields("wk7").Value
        week8a = rstSearch9.Fields("wk8").Value
        week9a = rstSearch9.Fields("wk9").Value
        week10a = rstSearch9.Fields("wk10").Value
        week11a = rstSearch9.Fields("wk11").Value
        week12a = rstSearch9.Fields("wk12").Value
        week13a = rstSearch9.Fields("wk13").Value
        week14a = rstSearch9.Fields("wk14").Value
        week15a = rstSearch9.Fields("wk15").Value
        week16a = rstSearch9.Fields("wk16").Value
        week17a = rstSearch9.Fields("wk17").Value
                Set rstSearch9 = nothing

%>
<!-- BEGIN HTML
------------------------------------------------------------ -->

<b><font size="5">Begin...ID = <%=rstSearch("ID")%><br>
</font></b><br>
Total qb= <%= datotal %>
<br>
RB1 total =<%= rb1_total%>
<br>
RB2 total = <%=rb2_total %>
<br>
WR1 total = <%=wr1_total %>
```

```
<br>
WR2 total = <%=wr2_total %>
<br>
K total = <%=k_total %><br>
<font size=2>
        <br><b><font color="#FF0000"><font color="black" size
="4"><%=rstSearch9.Fields("username").Value %></font>
        WEEK</font><font color="#00FF00" size="4"> <%=request.querystring("weekNum")%>
        </font><font color="#FF0000">TOTAL=</font></b><font color="#FF0000"><%= week_total
%></font></font><br><b><font size="4">END</font></b><hr>


<!-- END HTML------
-->

<%
'=============================================================================================
        rstSearch.MoveNext
Loop
'=============================================================================================
set rstSearch = nothing
%></font></table><!-- #include file="admin_nav.asp" -->
```

Mod.css File used to control the style of the pages:

```
body
        {
        color: black;
        font-family: Verdana, serif;
        background-color: silver;
        margin: 2.5%
        }
td.info
        {
        color: black;
        background-color: white;
        font-family: Verdana, serif;
        font-size: 11pt;
        }
td.score
        {
        color: red;
        background-color: white;
        font-family: Verdana, serif;
        font-size: 8pt;
        text-align: center;
        }

tr.0
        {
        background: #D7D7E3;
        font-family: Verdana, serif;
        color: black;
        font-weight: 300;
        font-size: 10pt;
        }

tr.1
        {
        background: white;
        font-family: Verdana, serif;
        color: black;
        font-weight: 300;
        font-size: 10pt;
        }

tr.2
        {
        background: #D7D7E3;
        font-family: Verdana, serif;
        color: black;
        font-weight: 300;
        font-size: 10pt;
        }

th
        {
        background: #6699FF;
        font-family: Verdana, serif;
```

```
        color:  white;
        font-weight: bold;
        font-size: 10pt;
        }

table
        {
        width: auto;
        border-style: hidden;
        }

input
        {
        color: white;
        font-family: Verdana, serif;
        font-weight: 700;
        background-color: #6699FF;
        }
hr.bottom
        {
        color: gray;
        width:  515px;
        background: silver;
        }

hr
        {
        color: #6699FF;
        background: white;
        }

select
        {
        font-family: Verdana, serif;
        font-weight: 700;
        background-color: #6699FF;
        width: auto;
        color: white;
        }
```